

平成25年度 修士論文

Wireless Body Area Networkのための  
XOR演算を用いた符号化

電気通信大学大学院 情報システム学研究科  
情報ネットワークシステム学専攻

学 籍 番 号:1252033

横田啓吾

指導教員

森田啓義 教授

加藤聰彦 教授

吉永努 教授

平成26年1月27日

# 目次

第1章	はじめに	3
1.1	研究背景	3
1.2	研究目的	4
1.3	研究方法	5
1.4	研究成果	5
1.5	本論文の構成	6
第2章	関連研究と提案手法	8
2.1	XOR演算を用いた符号化	8
2.2	既存研究	9
2.2.1	XOR符号化を適用するWBANのモデル	10
2.2.2	グラフ理論を用いたXOR符号化方式の評価	15
2.3	提案手法	21
2.3.1	XOR符号化の弱点	21
2.3.2	連接符号化の提案	22
第3章	提案手法における消費電力	26
3.1	FPGA設計による消費電力の推定	26
3.2	消費電力の検証	26
3.2.1	検証環境	26
3.2.2	符号器の設計	27
3.2.3	検証結果	29
第4章	XOR符号化方式の検証	30
4.1	XOR符号化方式の構成パターン検証	30
4.1.1	送信器の設定	30
4.1.2	検証するXOR符号化方式の構成パターン	31
4.1.3	検証環境	34

4.1.4	検証結果 . . . . .	36
4.1.5	考察 . . . . .	36
4.2	XOR符号化方式の総当り検証 . . . . .	37
4.2.1	検証準備 . . . . .	37
4.2.2	検証結果 . . . . .	41
4.2.3	考察 . . . . .	45
<b>第5章</b>	<b>提案手法における誤り耐性</b>	<b>47</b>
5.1	接続符号化の性能検証実験 . . . . .	47
5.1.1	検証準備 . . . . .	47
5.1.2	検証する伝送システム . . . . .	47
5.1.3	接続符号化の性能検証結果 . . . . .	53
5.1.4	考察 . . . . .	54
5.2	パケット数を少なくした場合の検証 . . . . .	55
5.2.1	用意するパケット数が4個の時のDEP . . . . .	55
5.2.2	用意するパケット数が5個の時のDEP . . . . .	57
5.2.3	用意するパケット数が6個の時のDEP . . . . .	58
5.2.4	考察 . . . . .	60
5.3	各タイプに必要な消費電力の見積もり . . . . .	60
5.4	復号できなかったパケットの個数 . . . . .	61
5.5	再送方式の検討 . . . . .	66
<b>第6章</b>	<b>まとめと今後の課題</b>	<b>68</b>
6.1	まとめ . . . . .	68
6.2	今後の課題 . . . . .	69

# 第1章 はじめに

## 1.1 研究背景

近年、技術の進歩により様々な電子機器の小型化が進んでおり、その中の1つとして無線機器が挙げられる。小型化された無線機器は、元々有線で通信を行っていた現場に導入されることによって、様々な利便性の向上に貢献している。例えば、手持ちのPCで有線LANからインターネットに接続しようとする、PCとルーターの間をケーブルで繋がらないといけないため、ケーブルの長さ分ではPCを持ち歩くことができなくなる。しかし、無線化されたLANであれば、PCとルーターの間をケーブルで繋ぐ必要がなくなり、通信可能範囲内であれば自由にPCを持ち運びができるため、手軽にインターネットに接続できるようになる。小型化された無線機が現場に導入されている中で、医療分野においては高齢化が進んでいる現状から特に大きな恩恵を受けている。

日本国内において、総人口に占める65歳以上の人口の割合（高齢化率）は、平成25年6月14日に閣議決定された最新の「高齢社会白書」によると平成24年10月1日現在で24.1%であり、今後も増え続けると予想されている[1]。高齢化が進むにつれて問題となるのが、医療現場で働く方々への負担である。高齢者の人口が増えると、必然的に病院を利用する人も増えていく。また、病院を利用する人が増えていくと、病院で働く職員への負担が重くなる。故に、高齢化率の上昇に備えて、患者が病院で受ける診察を支援するシステムの開発が求められる。

そこで、脈拍などの生体情報を取得するセンサーを無線経由で情報を送れるようにし、遠隔で患者の生体情報を常にリアルタイムで監視するシステム（見守りシステム）の研究・開発が進められている[2][3][4][5]。従来の有線のセンサーで常に監視する場合、データを受信する機器から患者が離れることが出来ないため、患者の行動に制限がかかっていた。センサーを無線化させると、通信可能範囲内であれば自由に動き回る事が



できるので、有線の時と比べて行動範囲が広がる。また、患者の家で用いることで、普通に生活しつつ生体データをリアルタイムで監視することも可能となる。よって、従来病院でしか受けることが出来なかった診察を家で受けることが可能になり、病院で働く職員の負担を減らすことができる。

また、このような見守りシステムで用いる無線通信は、Wireless Body Area Network (WBAN) と呼ばれている。WBANとは、体の表面や内部、及び近辺に無線機器を設置して行う近距離無線通信で、通信可能範囲はおよそ3[m]以内と比較的狭い範囲である。WBANは、医療や福祉などの現場で活用することが期待されている。このネットワークで用いる機器を人体に付ける、もしくは埋め込んで使用する用途上、その機器を小さくかつ軽くすることが求められる。よって、機器の電池も小さくなるため、必要な消費電力を抑えなければならない。また、先ほどの見守りシステムなど、医療の現場で用いる際は通信誤りが医療ミスに繋がる恐れがあるので、誤りをできるだけ生じないようにする必要がある。

## 1.2 研究目的

無線通信で用いられる符号化において、XOR演算のみを用いて、通信時に発生するパケット損失誤りを訂正可能な符号化法が提案されている[6][7]。この符号化では、計算が単純なXOR演算のみ使用しているため、従来の符号化と比べて計算量が低く、消費電力も少なくなると予想されるので、WBANへの適用が期待されている。しかしながら、この符号化はパケット損失誤りを訂正できるが、他の誤り（例えば、受信したパケット内に含まれうるビット反転誤り）は訂正できない。

そこで本研究では、このXOR演算を用いた符号化とビット反転誤りが訂正可能なBCH符号の符号化を接続させる接続符号化を提案することによって、WBANのための省電力かつパケット損失誤りとビット反転誤りに対して耐性の高い符号化法の考案を目指す。

## 1.3 研究方法

本研究では、次の3つの取り組みを行う。

- 提案する接続符号化の構成要素である XOR 演算を用いた符号化と BCH 符号の符号化について、符号化に必要な消費電力の見積りを行う。
- XOR 演算を用いた符号化方式について、すでに既存研究 [7] である方式が提案されているが、その性能がどれくらい優れているかは、符号化方式に対応したグラフ構造の場合の数が大きすぎるために十分な評価は行われていない。そこで、これまで見い出されている最良の符号化方式のグラフ構造に対して、他に構成可能な符号化方式のグラフ構造について評価することによって、より性能の優れた符号化方式の探索を行う。
- さらに、通信中に発生するパケット損失とビット反転誤りに対する耐性や消費電力の観点から、提案する接続符号化に対して検証を行う。

## 1.4 研究成果

まず、提案する接続符号化の構成要素である XOR 演算を用いた符号化と BCH 符号の符号化を FPGA アプリケーション開発に用いられる Verilog-HDL によって記述し、符号化にかかる消費電力の見積りを行った。

次に、一般に医療用 WBAN で用いられるセンサーの数は 10 個程度までとされている [9]。ここでは用いるセンサーの数を 9 と設定し、更に冗長度を 2（受信器へ送るパケット総数を 18）とした時に、既存研究 [7] における検証で最もパケット損失耐性が高かった XOR 符号化方式、及び別の XOR 符号化方式を 3 つ用意し、各方式におけるパケット損失耐性を、一定確率でパケット損失が発生する環境で複数回通信を行った際の通信失敗確率を求めて評価した。その結果、これら 4 つの XOR 符号化方式の中では、既存研究 [7] の方式が最もパケット損失耐性が高いことがわかった。さらに、符号化方式に対応したグラフ構造からパケット損失耐性を評価する検証方法を用いて、冗長度を 2 に固定した状態でセンサーの数を 4, 5, 6 個に減らした場合に、構成可能な全ての符号化方式に対応したグラフ構造を考えて全探索することによって、各センサーの数において最もパケット損失

に対する耐性が高い方式に対応したグラフ構造が得られた。さらに、得られたグラフ構造に対応した符号化方式は、既存研究[7]の提案アルゴリズムから構成される方式と必ず一致するとは限らないことがわかった。

また、提案する接続符号化において、パケット損失とビット反転誤りの両方が同時に発生する場合に対する耐性を検証した。ここでは、まずは送るべきパケットが復号時に全て復号できたかどうかから両方の誤りが同時に発生した場合に対する耐性を検証し、誤り訂正能力の評価を行った。その結果、各誤りの発生確率によって提案する接続符号化を用いる方が良い場合と用いない方が良い場合があることがわかった。さらに、XOR演算を用いた符号化を行った後にBCH(63,51)符号の符号化を行う符号化(タイプ2)と、これら2つの符号化を逆の手順で接続させた符号化(タイプ3)を比べたところ、ビット反転誤り発生確率が高くなるほど(確率が0.003以上になると)、タイプ2に比べてタイプ3の方がより高い誤り耐性を示すが、タイプ2とタイプ3で耐性にそれほど違いが出ない時(例えばビット反転誤りが発生する確率が0.001の時)にはタイプ2の方が、タイプ3に比べて符号化に必要な消費電力が少ないため、WBANの符号化として、より適していることがわかった。しかし、復号時に復号できなかったパケットの個数をから誤り訂正能力を評価したところ、パケット損失の確率が0.46以上になる時、BCH符号化単体で符号化を行ったほうが、タイプ2やタイプ3と比べて復号できなかったパケットの個数が少なくなることがわかった。

## 1.5 本論文の構成

本論文の構成は以下の通りである。

- 2章では、本研究で着目する符号化の仕組み、及びこの符号化に関する既存研究と提案手法の接続符号化について紹介する。
- 3章では、XOR演算を用いた符号化とBCH符号の符号化をFPGA設計を用いて電力消費の見積りを行う。
- 4章では、様々なXOR演算を用いた符号化の方式について、パケット損失に対する耐性の検証・比較を行う。

- 5章では，提案する連接符号化に対して，パケット損失とビット反転誤りの両方が同時に発生する場合に対する耐性の検証を行う．
- 6章では，本論文のまとめを行う．

## 第2章 関連研究と提案手法

本章では，本研究で着目するXOR演算を用いた符号化の仕組み，及びこの符号化に関する既存研究を紹介する．また，本研究で提案する接続符号化についても紹介する．

### 2.1 XOR演算を用いた符号化

低消費電力での使用できることが期待されている符号化の手法の1つとして，XOR演算を用いるものがある．XOR演算とは「0」または「1」を並べて構成されるビット列に対して行う演算であり，2つの同じ長さのビット列を入力として，同じ位置のビット毎に排他的論理和（XOR）を行って同じ長さのビット列を出力する演算であり，演算子は $\oplus$ で表される．つまり，各ビットの位置において入力された2つのビットが違う値であれば出力されるビットは「1」，同じ値であれば「0」となる．また，この演算は出力されるビット列と入力したビット列のうち1つをXOR演算すると，演算結果のビット列と入力したビット列の一方から他方が得られる．例えば，2つのビット列0101と1100をXOR演算すると， $0101 \oplus 1100 = 1001$ となる．また，得られたビット列1001と，入力したビット列のうちの1つの0101をXOR演算すると， $1001 \oplus 0101 = 1100$ となり，もう1つの入力したビット列を得ることができる．

本研究では，既存研究[6][7]で提案されている送信パケット同士をXOR演算することで得られるパケットを作成して，元の送信パケットと共に送信する方式に着目する．この方式では，データ通信中にパケットが損失した場合，冗長なパケットを用いることにより損失したパケットを復号することができる．この符号化を，以下では「**XOR**符号化」と呼ぶことにする．

例 1  $P_1$  と  $P_2$  の 2 つのパケットを送りたい場合、XOR 符号化を用いて次の手順で符号化と復号を行う。

1. パケット  $P_1$  と  $P_2$  をビット毎に XOR 演算することにより、パケット  $P_1 \oplus P_2$  を作成する。
2. 送信器から受信器へ、 $P_1$  と  $P_2$  および  $P_1 \oplus P_2$  の 3 つのパケットを送信する。
3. 受信器に届いたパケットの種類によって、次の作業を行う。
  - $P_1$  と  $P_2$  が受信器に届いた場合、送りたい 2 つのパケットは届いているので、冗長なパケットである  $P_1 \oplus P_2$  が受信器に届いていても復号処理を行わない。
  - $P_1$  のみが損失して  $P_2$  と  $P_1 \oplus P_2$  が受信器に届いた場合、次の演算を行うことにより  $P_1$  の復号を行う。

$$P_2 \oplus (P_1 \oplus P_2) = P_1 \quad (2.1)$$

- $P_2$  のみが損失して  $P_1$  と  $P_1 \oplus P_2$  が受信器に届いた場合、次の演算を行うことにより  $P_2$  の復号を行う。

$$P_1 \oplus (P_1 \oplus P_2) = P_2 \quad (2.2)$$

- 送信された 3 つのパケットのうち 2 つが損失して 1 つしか受信出来なかった場合、届いたパケットのみでは損失したパケットを復号できないので復号処理を行わない。
- 送信された 3 つのパケット全てが損失して何も受信できなかった場合、復号できないので復号処理は行わない。

## 2.2 既存研究

本節では、XOR 符号化を WBAN へ適用するモデル、及び XOR 符号化方式（どのパケット同士を XOR 演算するかを決める構成）を提案した既存研究 [6][7] について紹介する。さらに、この XOR 符号化方式に対して、グラフ理論を用いて通信におけるパケット損失に対する耐性について調査した既存研究 [7] についても紹介する。

### 2.2.1 XOR符号化を適用するWBANのモデル

本研究では、図2.1のような、ある患者の脳波や心電図などの生体情報を医者がリアルタイムで監視するWBANのモデルを想定する。まず、センサーを患者の皮膚上に装着したり体内に埋め込むことで、脳波や心電図などの情報を得る。医療用のWBANに用いるセンサーの数は、一般的に10個程度までとされている[9]。

センサーからの情報は、配線を通して患者の体に装着している送信器へ送られる。ここで、各センサーが送る情報は、パケットと呼ぶことにする。

また、送信器では必要に応じて受け取ったパケットに符号化を行い、観測局 (MS) へ送られる。送信器で行われる符号化では、各センサーからのパケットを入力として、その中から2つのパケットを選んでXOR演算を行ったパケットが出力されるが、XOR演算を行わずにそのまま出力されるパケットもある。どのパケット同士をXOR演算して出力するのか、もしくはどのパケットをXOR演算せずに出力するのかは、XOR符号化方式として通信を行う前に決めておく。

さらに、MSでは送信器からパケットを受け取り、必要に応じて復号を行い、センサーから送られてきたパケットに基づいて患者の生体情報を表示する。

ここでは、WBANの無線通信は送信器からMSまでの通信で用いられ、センサーから送信器までの通信は一切雑音が発生せず正確な情報を送れていると仮定する。

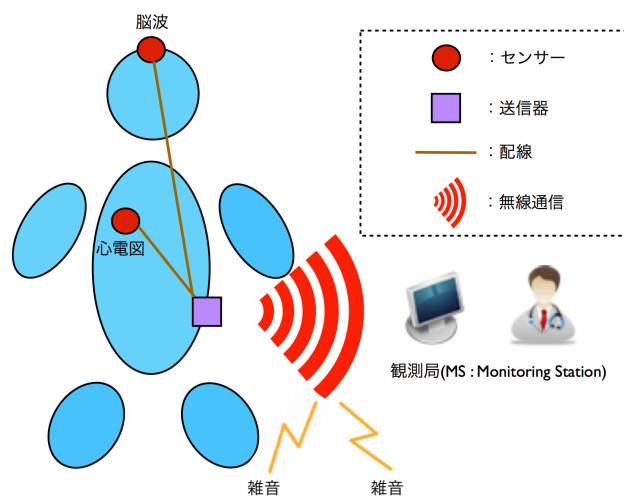


図 2.1: 患者の生体情報を医者がリアルタイムで監視しているモデル

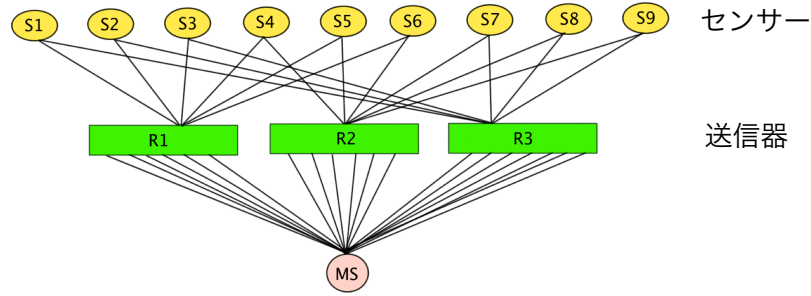


図 2.2: 機器間のパケット通信の図式化

機器間のパケット通信を図式化したものを，図2.2に示す．この図の場合，患者に用いるセンサーの数は9個，送信器の数は3個と想定し， $S_i (i = 1, 2, \dots, 9)$  と  $R_j (j = 1, 2, 3)$  はそれぞれセンサーと送信器に対応している．また，センサー  $S_i$  から送られるパケットを  $P_i$  とする．

センサー  $S_i$  と送信器  $R_j$  を繋ぐ線は，センサー  $S_i$  から送信器  $R_j$  へパケット  $P_i$  が送られることを表している．なお，各センサーから同じパケットを送信器へ複数個送ることに注意されたい．また，この各センサーから送られるパケットの数を「冗長度」と呼ぶことにする．この図では，冗長度は2としている．例えば， $S_1$  は  $R_1$ ， $R_3$  と線で繋がれているので，センサー  $S_1$  からパケット  $P_1$  が送信器  $R_1$  と  $R_3$  へ送られていることがわかる．

また，各送信器に入力されるパケットの数と，各送信器から出力されるパケットの数は同じであると仮定する．つまり，全ての送信器に入力されるパケットの総数と，全ての送信器から出力されるパケットの総数は同じとなる．よって，センサーの数と冗長度が決まると，必然的に各送信器から出力されるパケットの総数，つまり（パケット損失が発生しなかった場合に）MSに送られるパケットの総数が決まる．図2.2の場合，センサーの数は9，冗長度は2なので，MSに送られるパケットの総数は  $9 \times 2 = 18$  個である．

さらに，送信器  $R_j$  と MS を繋ぐ線は，送信器  $R_j$  から MS へあるパケットが送られることを表している．なお，ここで送られるパケットには必要に応じて符号化されたパケットを含んでいる．また，通信中にパケットが損失した場合は，そのパケットに対応する線が消えるとする．例えば，送信器  $R_1$  から MS に送信されたパケットのうち1つが通信中に損失した場合，そのパケットに対応する線が図2.3における赤線だとすると， $R_1$  から



MSに繋がっている赤線が図から消失する。

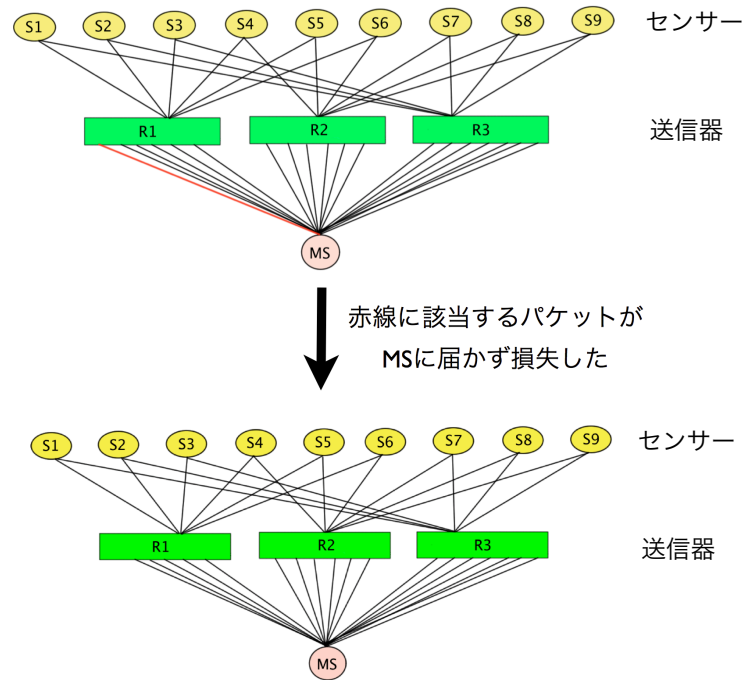


図 2.2 の場合において，各パケット  $P_i$  がどの送信器  $R_j$  に送られるかをまとめると表 2.1 となる．

表 2.1: 各送信器に届くパケット

送信器	センサーから送信器へ送られてくるパケット					
$R_1$	$P_1$	$P_2$	$P_3$	$P_4$	$P_5$	$P_6$
$R_2$	$P_4$	$P_5$	$P_6$	$P_7$	$P_8$	$P_9$
$R_3$	$P_7$	$P_8$	$P_9$	$P_1$	$P_2$	$P_3$

これらのパケットは，必要に応じて符号化して MS へ送られる．ここでは，各送信器で表 2.2 となるように符号化を行い MS へ送信したとする．なお，この XOR 符号化方式は既存研究 [7] で紹介されているアルゴリズムに基づいている．このアルゴリズムでは XOR 演算しないパケットの個数を任意に設定する．この個数によって通信中に発生するパケット損失に対する耐性が変化するが，この事項については次節 2.2.2 で述べる．

表 2.2: 送信器から MS へ送られるパケット

送信器	送信器から MS へ送られるパケット					
$R_1$	$P_1$	$P_2$	$P_3 \oplus P_4$	$P_4 \oplus P_5$	$P_5 \oplus P_6$	$P_6 \oplus P_1$
$R_2$	$P_4$	$P_5 \oplus P_8$	$P_6 \oplus P_7$	$P_7 \oplus P_8$	$P_8 \oplus P_9$	$P_9 \oplus P_4$
$R_3$	$P_7$	$P_8 \oplus P_2$	$P_9 \oplus P_1$	$P_1 \oplus P_2$	$P_2 \oplus P_3$	$P_3 \oplus P_7$

パケットの復号方法の例を以下に示す.

例 2 MS 側において, 通信中にいくつか損失して表 2.3 のようなパケットを受信できたとする.

表 2.3: 各送信器から MS が受信できたパケット

送信器	MS が受信できたパケット					
$R_1$	$P_1$	$P_2$		$P_4 \oplus P_5$	$P_5 \oplus P_6$	
$R_2$			$P_6 \oplus P_7$	$P_7 \oplus P_8$		$P_9 \oplus P_4$
$R_3$	$P_7$	$P_8 \oplus P_2$	$P_9 \oplus P_1$	$P_1 \oplus P_2$	$P_2 \oplus P_3$	

これらのパケットを利用して, 例えば次の手順で復号ができる. ただし, MS は受信できたパケットの種類を判別できているものとする.

### 手順 1

何も復号を行っていない段階では, センサーから送信器へ送られたパケット  $P_1, P_2, \dots, P_9$  の中で受信できたパケットは表 2.3 より  $P_1, P_2, P_7$  である. このうち  $P_2$  と  $P_7$ , 及び受信できた XOR 演算済みのパケットを利用することで, 次の XOR 演算を行うことで 3 つのパケットが復号できる.

$$(P_2 \oplus P_3) \oplus P_2 = P_3$$

$$(P_6 \oplus P_7) \oplus P_7 = P_6$$

$$(P_8 \oplus P_2) \oplus P_2 = P_8$$

ここまで復号できたパケットをまとめると表 2.4 のようになる. ただし, 黒字のパケットは現在の手順の前に復号, もしくは送信器から受信できたパケットを表す. また, 赤字のパケットは現在の手順で復号できたパケットを表す.

表 2.4: 手順 1 までに復号できたパケット

復号できたパケット							
$P_1$	$P_2$	$P_3$			$P_6$	$P_7$	$P_8$

## 手順2

表2.4の復号できた $P_6$ と表2.3の $P_5 \oplus P_6$ を用いると,

$$(P_5 \oplus P_6) \oplus P_6 = P_5$$

を復号できる. ここまでをまとめると表2.5となる.

表 2.5: 手順2までに復号できたパケット

復号できたパケット								
$P_1$	$P_2$	$P_3$		$P_5$	$P_6$	$P_7$	$P_8$	

## 手順3

表2.5の復号できた $P_5$ と表2.3の $P_4 \oplus P_5$ を用いると,

$$(P_4 \oplus P_5) \oplus P_5 = P_4$$

を復号できる. ここまでをまとめると表2.6となる.

表 2.6: 手順3までに復号できたパケット

復号できたパケット								
$P_1$	$P_2$	$P_3$	$P_4$	$P_5$	$P_6$	$P_7$	$P_8$	

## 手順4

表2.6の復号できた $P_4$ と表2.3の $P_9 \oplus P_4$ を用いると,

$$(P_9 \oplus P_4) \oplus P_4 = P_9$$

を復号できる. ここまでをまとめると表2.7となる.

表 2.7: 手順4までに復号できたパケット

復号できたパケット								
$P_1$	$P_2$	$P_3$	$P_4$	$P_5$	$P_6$	$P_7$	$P_8$	$P_9$

表2.7より $P_1$ から $P_9$ まで復号できたので, これ以上の復号は行わない.

例2のように, 復号においては受信できたパケットのうちXOR演算されていないパケットとXOR演算されているパケット同士のXOR演算を行う. これにより, 送信器からMSに送られるべきパケットが全て受信されなくとも, 符号化で生成したXOR演算されているパケットを用いること

でセンサーから送信されたパケットを復号することができる。ただし、MSに届くパケットの種類によってはセンサーから送信されたパケットを全て復号できないことがあるので注意されたい。

## 2.2.2 グラフ理論を用いたXOR符号化方式の評価

### XOR符号化方式のグラフ表記

これまで紹介してきたXOR符号化方式について、既存研究[7]ではグラフ理論を用いて通信中のパケット損失に対する耐性を調査している。ここでは、その既存研究で用いられた方式のグラフ表記について説明する。

XOR符号化方式をグラフ表記するには、以下の手順で行う。

手順1：各センサーから送られるパケット  $P_i$  に対応する頂点を用意する。

手順2：XOR符号化の際にXOR演算されなかったパケットに対応する頂点に自己ループ（以降、ループと記す）を付ける。

手順3：XOR演算したパケットが存在すれば、そのパケットを構成する元となった2つのパケットに対応する頂点同士を辺で繋ぐ。

ここで、表2.2の方式を例にとって説明する。

例3 表2.2では、9つのセンサーからそれぞれ  $P_1, P_2, \dots, P_9$  が送られているので、手順1より図2.4のように9つの頂点が用いられる。

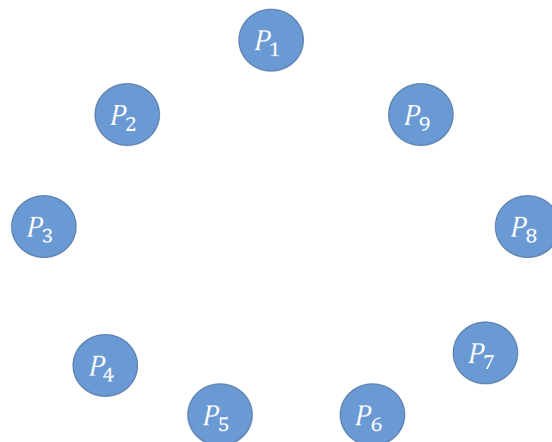


図 2.4: 各  $P_i$  に対応する頂点を配置

表 2.2 において,  $P_1, P_2, P_4, P_7$  が XOR 演算されていないので, 手順 2 より 図 2.5 のようにこれらのパケットに該当する頂点にループを付ける.

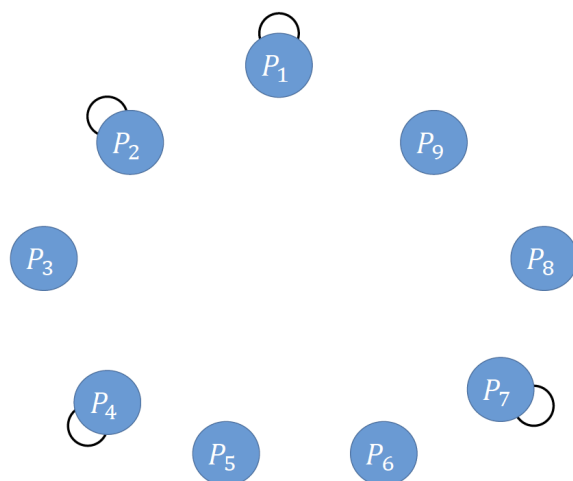


図 2.5: ループを配置

さらに, 表 2.2 には  $P_3 \oplus P_4$  が存在するので, 手順 3 より  $P_3$  と  $P_4$  に対応する頂点同士を辺で繋ぐことになる. 他のパケットに対しても同様に辺を繋ぐと, 図 2.6 のようになる.

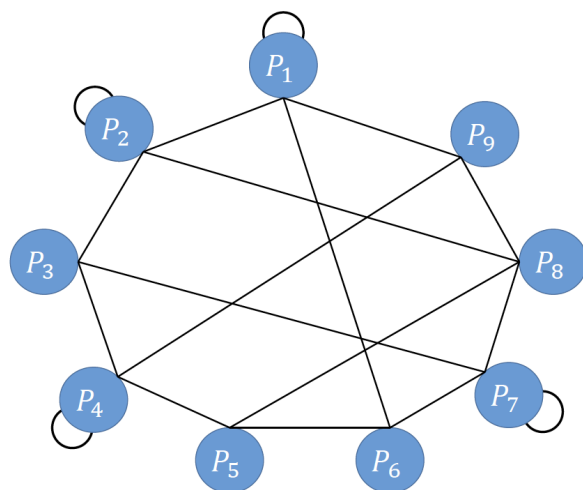


図 2.6: 辺を配置

このグラフ表記では送信器から MS までの通信中にパケットが損失した場合, そのパケットに対応する辺を取り除く. 例えば, 図 2.6 から  $P_1$  が損

失した場合は図2.7,  $P_3 \oplus P_4$ が損失した場合は図2.8のようになる.

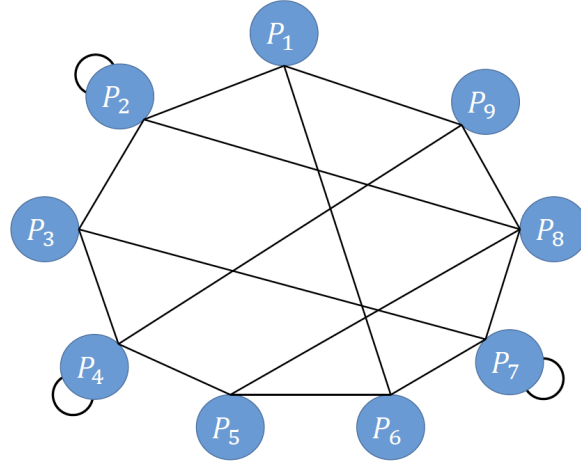


図 2.7: 通信中に  $P_1$  のパケットが損失

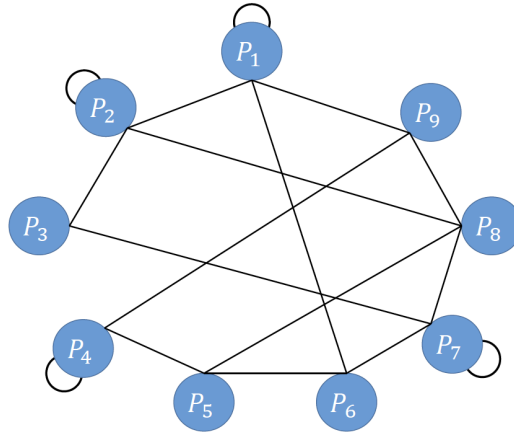


図 2.8: 通信中に  $P_3 \oplus P_4$  のパケットが損失

このグラフ表記において, 次の定理が成り立つ.

**定理 1 (Byrne ら [7])** XOR 符号化方式をグラフ表記したものを  $G$ , また通信中に損失したパケットに対応する辺を取り除いてできる  $G$  の部分グラフを  $H$  とする. グラフ  $H$  の各成分が 1 つ以上のループを持つ時, 全てのパケットを復号することができ, またその時に限る.

あるグラフが全てのパケットを復号することができる状態を復号可能と呼び, そのようなグラフを復号可能なグラフを呼ぶ. また,  $n$  個の頂点と  $m$  個の辺を持つ復号可能なグラフの集合を  $D(n, m)$  で表す.

例 4 3つのパケット  $P_1, P_2, P_3$  において図2.9の3つのグラフを考える.

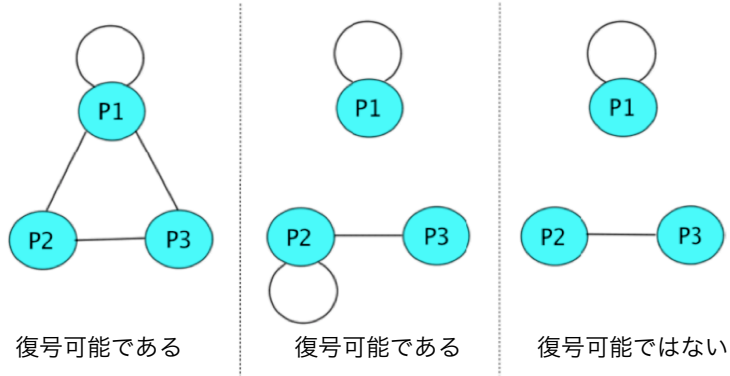


図 2.9: 復号可能か復号可能でないかの例

左のグラフは、頂点  $P_1, P_2, P_3$  が辺で繋がっているため、全体で1つの成分を構成している。また、この成分はループを1つ持っているので、これは復号可能である。中央のグラフは、 $P_1$  が  $P_2, P_3$  と辺で繋がっていないため、 $P_1$  または  $P_2, P_3$  で2つの成分を構成している。しかし、2つの成分はそれぞれループを持っているので、これも復号可能である。右のグラフは中央のグラフと同様に2つの成分を構成しているが、 $P_2$  と  $P_3$  で構成される成分はループを持っていないので復号可能ではない。

## XOR 符号化方式の評価

ある XOR 符号化方式において、通信中に発生するパケット損失に対する耐性を評価するため、次の定義を用いる。

**定義 1 (Byrne ら [7])** グラフ  $G \in D(n, m)$  において、復号可能である確率 (Decoding Probability)  $P_G$  を

$$P_G = \sum_{x=0}^m C_x^G q^{m-x} (1-q)^x \quad (2.3)$$

で定義する。ただし、 $C_x^G$  は  $G$  から  $x$  本の辺を取り除いた際にできる復号可能なグラフの個数、また  $q$  は辺が取り除かれない確率とする。

このグラフ表記では通信中にパケットが損失した際に，それに該当する辺をグラフから取り除くことを行う．つまり，定義1の $q$ は辺が取り除かれない確率であるが，これは通信中に各パケットが消失する確率がそれぞれ $q$ であることに等しい．故に，あるXOR符号化方式のパケット損失に対する耐性は，この方式に対応するグラフ $G$ について適当な $q$ を設定して式(2.3)を用いることで評価できる．

例 5 3つのパケット $P_1, P_2, P_3$ を用いた図2.10のようなグラフ $G$ を考える．この $G$ において， $x$ 本の辺を取り除いた際にできる復号可能なグラフの個数 $C_x^G$ を調べる．

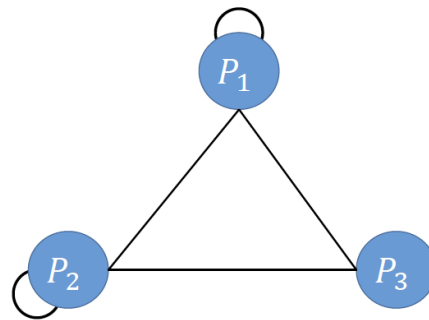


図 2.10: グラフ  $G$

#### $x = 0$ の時

$x = 0$ ，つまりグラフ $G$ から0本の辺を取り除いた際にできる復号可能なグラフの個数が $C_0^G$ である．グラフ $G$ から0本辺を取り除くとは，このグラフから辺を何も取り除かないことを意味する．故にグラフ $G$ 自身が復号可能であるのかを調べるが，図2.10より頂点 $P_1, P_2, P_3$ がそれぞれ辺で繋がっていて1つの成分を構成しており，かつループを持っているので復号である．つまり， $C_0^G = 1$ である．

#### $x = 1$ の時

$x = 1$ ，つまりグラフ $G$ から1本の辺を取り除いてできる復号可能なグラフの個数が $C_1^G$ である．グラフ $G$ から辺を1本取り除いて出来るグラフの個数は，グラフの辺から1本の辺を選ぶ個数と等しいので ${}_5C_1 = 5$ 個存



在する．これら5個のグラフは，図2.11のようになる．さらに，これら5個のグラフは全て復号可能なグラフであるので， $C_1^G = {}_5C_1 = 5$ である．

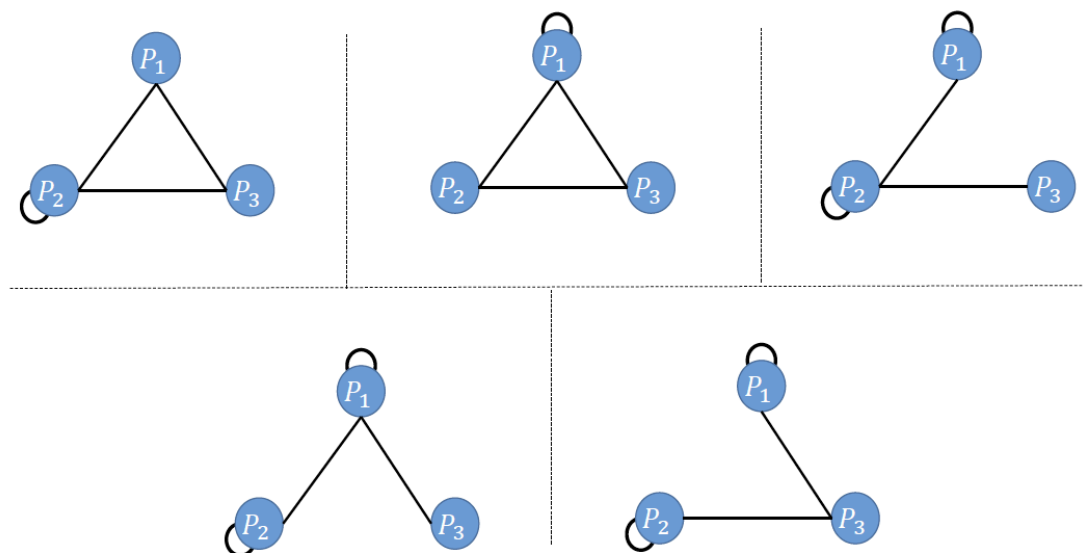


図 2.11: グラフ  $G$  から辺を1本取り除いてできるグラフ

#### $x = 2$ の時

グラフ  $G$  から辺を2本取り取り除いて出来るグラフの個数は， $x = 1$ の時と同様に  ${}_5C_2 = 10$  個存在する．これら10個のグラフについて調べると，図2.12の2つのグラフのみ復号可能でないことがわかる．つまり，この2つのグラフを除くグラフが復号可能であるので， $C_2^G = {}_5C_2 - 2 = 10 - 2 = 8$  となる．

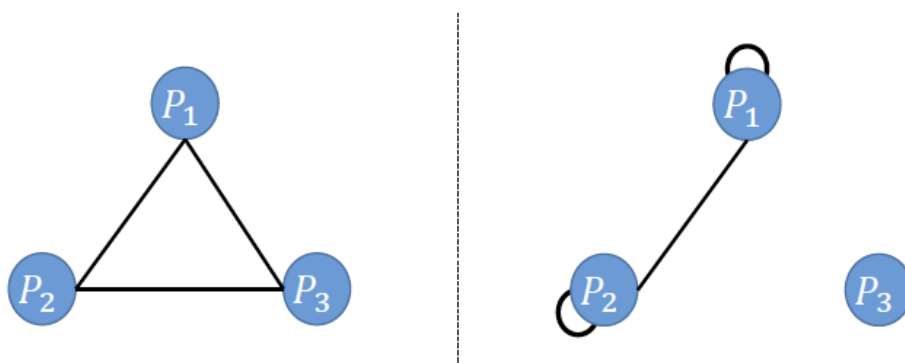


図 2.12: グラフ  $G$  から辺を2本取り除いてできるグラフのうち，復号可能でないグラフ

### $3 \leq x \leq 5$ の時

$x$ が3個以上の場合、辺を取り除いてできる全てのグラフは復号可能でないため、 $C_3^G = C_4^G = C_5^G = 0$ となる。

以上をまとめると、図2.10のグラフ $G$ と $C_x^G$ の関係性は表2.8となる。

表 2.8: 図 2.10 のグラフ  $G$  と  $C_x^G$  の関係性

$x$	0	1	2	3	4	5
$C_x^G$	1	5	8	0	0	0

XOR符号化では、表2.2のようにいくつかXOR演算せずにMSへ送るパケットがある。この個数は方式を構成する既存研究のアルゴリズムを用いる際に任意に設定できるが、その個数によってパケット損失に対する耐性が変化する。これについて、既存研究[7]では次の証明がされた。

**定理 2 (Byrneら [7])** XOR符号化方式における冗長度を $r$ と置く。また、ある符号化方式をグラフ表記したグラフを $G$ と置く。今、 $G$ において、 $2r-1$ 本の辺を取り除くことで初めて復号可能でない $G$ の部分グラフが現れると仮定する。この時、 $G$ に含まれるループの数 $L_G$ が $2r-1 \leq L_G \leq 2r$ の時に $C_{2r-1}^G$ が最大になる。

これより、XOR演算せずに送るパケットの個数を $2r-1$ もしくは $2r$ とする方式を用いると、パケット損失に対する耐性が高くなることが期待される。

## 2.3 提案手法

### 2.3.1 XOR符号化の弱点

XOR符号化は、ビット反転誤り (Bit Flip Error, BFE) といったパケットの損失以外の誤りを修復することができない。BFEとは、通信中にパケット内のビットデータが反転する (0が1へ、1が0へ反転する) 誤りのことである。

例 6 4-bit のデータから構成されているパケット  $P_1 = (1, 0, 0, 1)$  と  $P_2 = (0, 0, 1, 1)$  を考える．これら 2 つのパケットを XOR 演算すると，

$$P_1 \oplus P_2 = (1, 0, 0, 1) \oplus (0, 0, 1, 1) = (1, 0, 1, 0)$$

となる．

ここで， $P_1$  の 1 番目，及び  $P_2$  の 2 番目のビットに BFE が発生したパケット  $\hat{P}_1 = (\underline{0}, 0, 0, 1)$  と  $\hat{P}_2 = (0, \underline{1}, 1, 1)$  を考える．これら 2 つのパケットを XOR 演算すると，

$$\hat{P}_1 \oplus \hat{P}_2 = (\underline{0}, 0, 0, 1) \oplus (0, \underline{1}, 1, 1) = (\underline{0}, \underline{1}, 1, 0)$$

となる． $\hat{P}_1 \oplus \hat{P}_2$  と  $P_1 \oplus P_2$  を比べると，1 番目と 2 番目のビットがそれぞれ反転していることがわかる．

この現象を，XOR 符号の復号に当てはめて考えてみる．XOR 符号の復号では，通信中に損失しなかったパケットを XOR 演算することで損失したパケットを復号する．しかしながら，例えば損失しなかった各パケットに 1-bit ずつ異なる場所に BFE が含まれていたとすると，復号で XOR 演算をすることで上記のように 2-bit の BFE を含んだパケットが出来上がってしまう．つまり，BFE を増大させてしまう可能性がある．

### 2.3.2 接続符号化の提案

従来手法の XOR 符号化は，通信中に発生する誤りはパケット損失のみと仮定している．通信距離が短い WBAN では，一般に通信路を人体が遮るシャドーイングや，建物の壁で電波が反射して生じるフェージングによる信号の受信レベルの低下からパケット損失が発生することが問題視されているが，無線通信を行う場所によっては，パケット損失以外にも BFE のような誤りが発生すると考えられる．そのため，XOR 符号化のみを用いると，パケット損失を復号できても BFE などの誤りを訂正することができないため，正確なデータのやり取りが行えない可能性がある．

また，IEEE における標準規格では，WBAN の通信に BCH 符号の符号化を用いている [8]．この符号化は，符号長や訂正能力の選択の自由度が大きいことや，一般に符号器や復号器の構成が簡易であることから，主に移動体通信で利用されており，BFE を訂正することができる．

そこで、本研究ではパケット損失訂正が可能な51-bit XOR符号化、もしくは63-bit XOR符号化と、BFEの訂正が可能なBCH符号の符号化を接続させる符号化を提案する[10].

### 51-bit XOR符号化

51-bit XOR符号化とは、51-bitのデータを持つパケット同士をXOR符号化する符号化である.

例えば、以下の2つのパケットを考える.

$$\begin{aligned} P_1 &= (1, 0, 1, 1, 0, 1, 0, 1, 1, 1, 0, \dots, 0, 1, 1, 0, 0, 1, 1, 0, 0, 0) \\ P_2 &= (\underbrace{0, 1, 1, 1, 1, 0, 0, 0, 1, 1, 1, \dots, 1, 0, 1, 0, 1, 0, 1, 1, 0, 1}_{51\text{-bit}}) \end{aligned}$$

この2つのパケットに対して、51-bit XOR符号化では $P_1$ と $P_2$ をXOR演算を行い、パケット $P_1 \oplus P_2$ を得る.

$$P_1 \oplus P_2 = (\underbrace{1, 1, 0, 0, 1, 1, 0, 1, 0, 0, 1, \dots, 1, 1, 0, 0, 1, 1, 0, 1, 0, 1}_{51\text{-bit}})$$

### 63-bit XOR符号化

63-bit XOR符号化とは、63-bitのデータを持つパケット同士を51-bitの時と同様にXOR符号化する符号化である.

### BCH(63,51)符号化

BCH(63,51)符号とは、51-bitのデータに対して12-bitのパリティ検査ビットを付加することにより、2-bitまで発生するランダム誤り、及び3-bitまでのバースト誤りを訂正することができる2元線形符号であり、IEEEではこの符号化をWBANへ適用することを標準規格としている[8]. ランダム誤りとは、データを構成するビット列において個々のビットに独立して発生する誤りのことである. また、バースト誤りとは、ビット列において連続した複数ビットに発生する誤りのことである. この符号を用いた符号化を、本研究では「**BCH(63,51)**符号化」と呼ぶことにする.

パリティ検査ビットを付加した後の63-bitのデータを求めるには、この符号に該当する $51 \times 63$ の生成行列 $G$ を用いて行う. この生成行列 $G$ は、 $51 \times 51$ の単位行列を $I$ 、 $51 \times 12$ の0,1成分からなる行列を $A$ とすると

$$G = (I|A) \tag{2.4}$$

で表せる. ここで、符号化前の51-bitのデータを成分とする $1 \times 51$ のベクトルを $\mathbf{u}$ とすると、パリティ検査ビットが付加された符号化後の63-bitの

データを成分とする  $1 \times 63$  のベクトル  $\mathbf{c}$  は

$$\mathbf{c} = \mathbf{u}G \pmod{2} \quad (2.5)$$

で求められる。

なお、この符号化は図2.2における送信器で行われるとする。また、接続符号化の構成は次の2種類とする。それぞれの構成を、ある2つの51-bitのパケットが入力され、それらを用いて接続符号化が行われる例で説明する。

1つ目は、図2.13のように入力された2つの51-bitのパケットそれぞれに BCH(63,51) 符号化を行い、それぞれ12-bitのパリティ検査ビットが付加された63-bitのパケットにする。その後、2つのパケット同士に対して63-bit XOR 符号化を行い、63-bitの符号化済みのパケットが出力する。

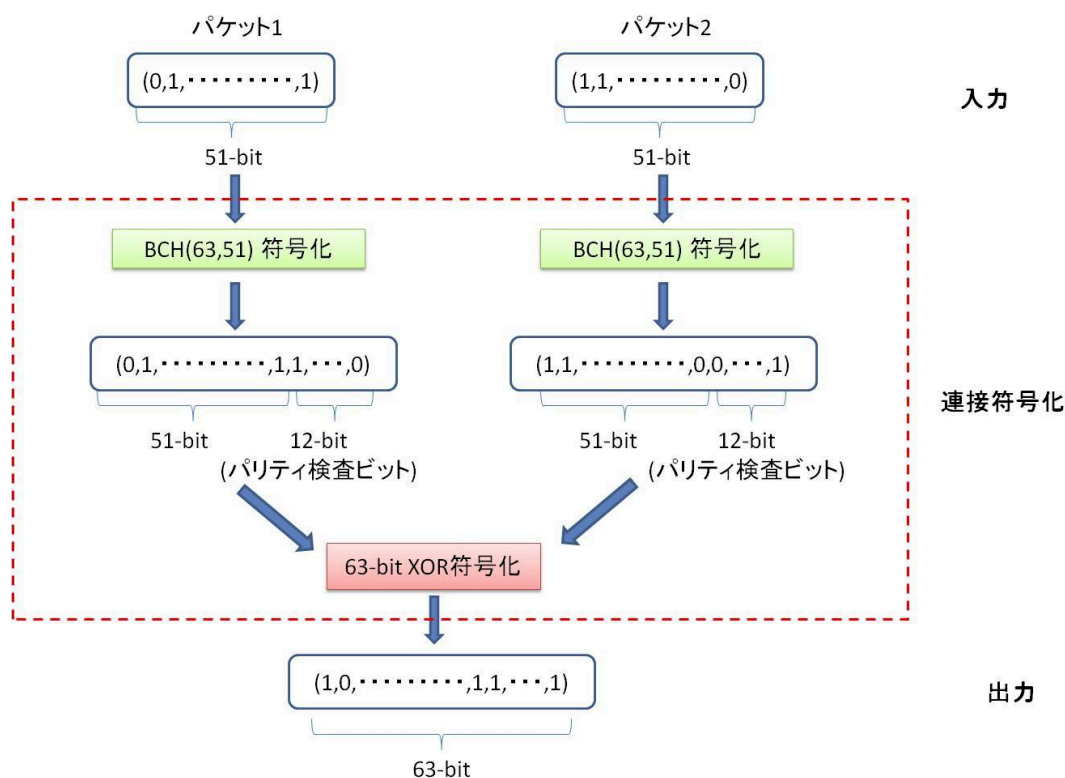


図 2.13: BCH→XOR 接続符号化 (5 章のタイプ 2 に対応)

2つ目は、図2.14のように入力された2つの51-bitのパケット同士に対して51-bit XOR 符号化を行い、そのXOR 符号化で得られたパケットに対し

てBCH(63,51)符号化を行うことで12-bitのパリティ検査ビットが付加された63-bitのパケットにして出力する。

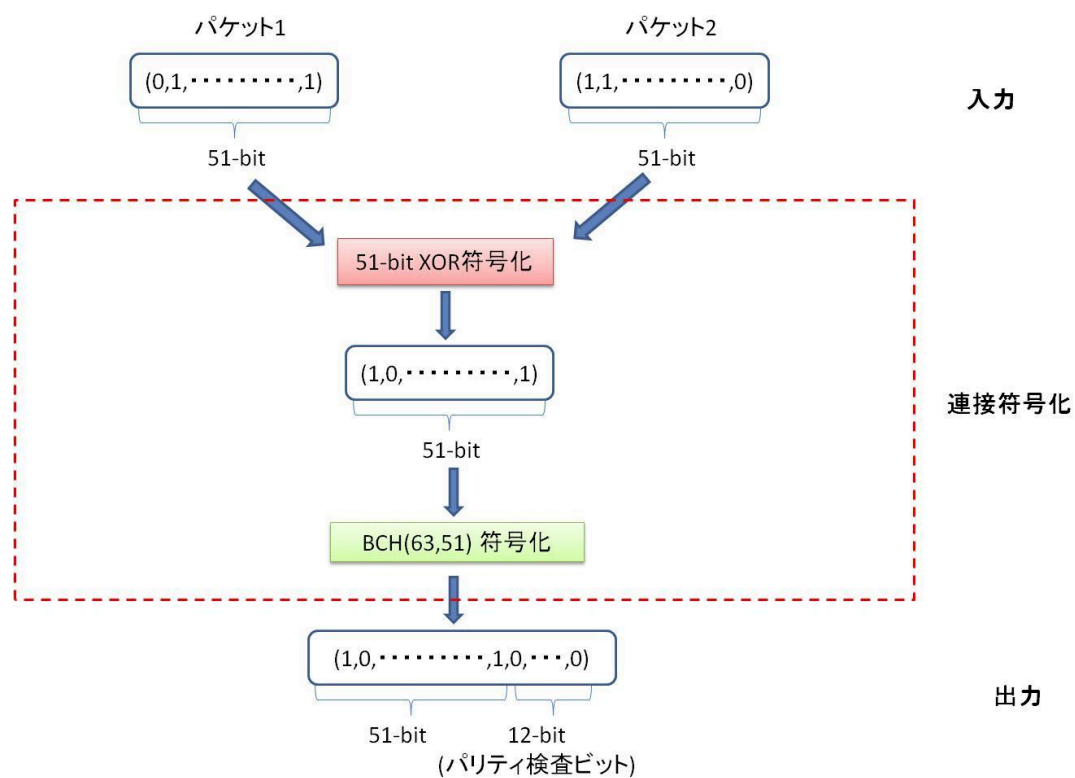


図 2.14: XOR→BCH 接続符号化 (5 章のタイプ 3 に対応)

この提案手法を用いることにより，通信中にパケット損失だけでなく BFE が発生した場合も，データを正確に受信することが可能になると考えられる。

## 第3章 提案手法における消費電力

本章では，提案する接続符号化の構成要素であるXOR演算を用いた符号化とBCH符号の符号化をFPGA設計を用いて電力消費を見積りを行う．

### 3.1 FPGA設計による消費電力の推定

符号化に必要な消費電力を測定する際は，その符号化方式が実装された無線機器のハードウェア媒体を用意して測定することが理想だが，実際に符号化部分のみを取り出して電力測定することは困難である．

一方，FPGAではハードウェア言語で記述することにより任意の回路を組むことが可能である．さらに，FPGA設計の際に用いるデザインツールには消費電力のシミュレータが付属されており，このシミュレータを用いることにより設計した任意の回路に必要な消費電力を見積もることができる．つまり，FPGAに符号化を行う回路を設計してシミュレータを用いることにより，符号化に必要な消費電力の見積りを行うことが可能である．この方法を，提案手法の接続符号化の構成要素であるBCH(63,51)符号化，51-bit XOR符号化，63-bit XOR符号化のそれぞれに対して用い，各符号化に必要な消費電力の見積りを行った．

### 3.2 消費電力の検証

#### 3.2.1 検証環境

各符号化をFPGAへ設計するにあたり，図3.1のような回路をハードウェア記述言語Verilog-HDLを用いて作成した．この回路内の各モジュールは，次のような働きをする．

**Interface** … 外からの情報データを符号器へ送り，符号化済みのデータを外へ渡す部分

**Encoder(符号器)** … 外からの情報データを符号化する部分（今回は，ここにBCHもしくはXORの符号化を設計する）

**State\_machine** … データの受け渡しのタイミングを管理する部分

作成した回路はデザインツールのXilinx ISE WebPACK 14.2を用いて論理合成を行い，このデザインツールに付属しているXPower Analyzerを用いて符号器部分の消費電力を検証した．なお，デバイスはXilinx Vertex-5 XC5VLX50，動作周波数は25.0[MHz]に設定した．

本研究において，各符号器をFPGAに設計するにあたり処理時間を揃えるため，符号化を行うデータの入力から出力までを1クロックサイクルで終わることができる設計を行った．各符号器における設計については次節で述べる．

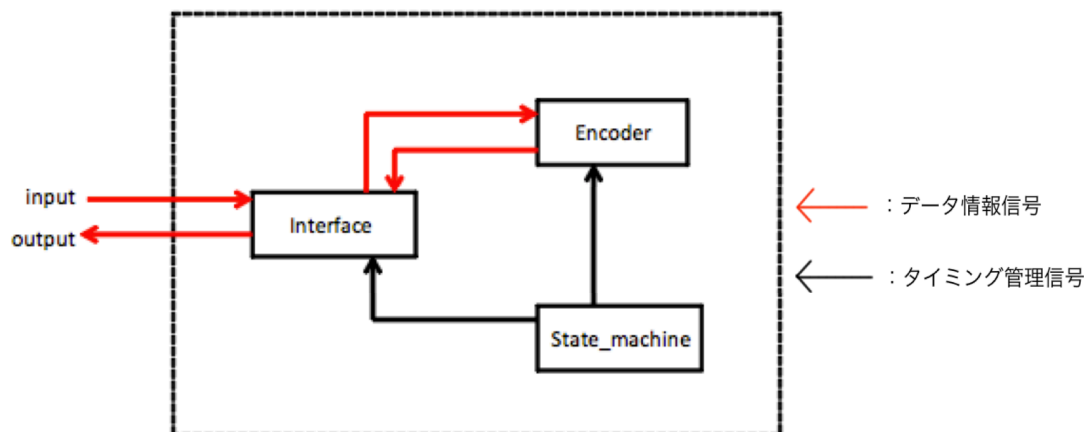


図 3.1: 設計したFPGA内の回路

### 3.2.2 符号器の設計

#### BCH(63,51) 符号器

符号化後の63-bitのデータを成分とするベクトル $\mathbf{c}$ は式(2.5)から求めることができるが， $\mathbf{c}$ を求めるには $\mathbf{c}$ に含まれる12個のパリティ検査ビット $p_1, p_2, \dots, p_{12}$ を計算すれば十分である．

それぞれのパリティ検査ビット $p_i$ は，行列 $A$ 内の $i$ 番目の列成分を $\mathbf{a}_i (i = 1, 2, \dots, 12)$ とおくと

$$p_i = \mathbf{u} \cdot \mathbf{a}_i \pmod{2} \quad (3.1)$$



で求めることができる。これより、図3.2のように式(3.1)における各 $i$ の $\mathbf{u} \cdot \mathbf{a}_i$ を計算する回路を作成し、並列に並べて同時に処理を行うと、 $\mathbf{u}$ のデータが入力されてから1クロックサイクル内で各パリティ検査ビット $p_i$ を出力させることができる。

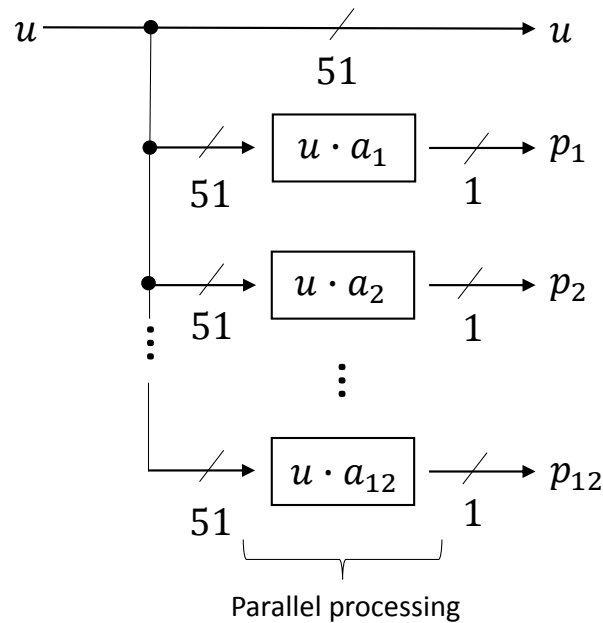


図 3.2: The BCH(63,51) 符号器

### 51-bit XOR 符号器

図3.3のように、2つの51-bitのデータを同時に入力し51-bit同士をXOR演算処理する回路に送ることで、1クロックサイクル内で2つの入力データをXOR演算したデータを出力させることができる。

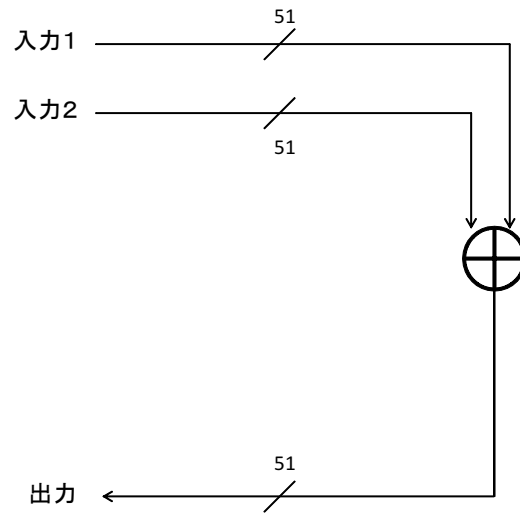


図 3.3: 51-bit XOR 符号器

### 63-bit XOR 符号器

51-bit XOR 符号器と同様に，63-bit 同士を XOR 演算処理する回路に 2 つの入力データを同時に入力して，入力から 1 クロックサイクル内で出力データを得ることができる．

### 3.2.3 検証結果

各符号化に必要な消費電力は，表 3.1 のようになった．なお，この結果はランダムな入力データの符号化を 10 万回行なった際の 1 クロックサイクル，つまり 1 回の符号化における平均消費電力である．

表 3.1: 1 回の符号化に必要な消費電力

	BCH(63,51)	51-bit XOR	63-bit XOR
Power[mW/clock cycle]	0.74	0.04	0.05

## 第4章 XOR符号化方式の検証

本章では，提案手法の構成要素の1つであるXOR符号化において，様々な方式についてパケット損失に対する耐性の検証・比較を行う．

### 4.1 XOR符号化方式の構成パターン検証

本論文で提案する連接符号化の構成要素の1つであるXOR符号化は，パケット損失が生じた時の訂正を行うために，どのパケット同士をXOR演算するのかを決めるXOR符号化方式を定める必要がある．既存研究[7]により提案されたアルゴリズムで構成されているXOR符号化方式と，パケット損失に対する耐性が高いことが示されている．しかし，このアルゴリズムが構成する方式が最も耐性が高いという保証は与えられていない．そこで，既存研究[7]のアルゴリズムから構成されたXOR符号化方式（パターン1），その符号化方式においてXOR演算を行うパケットの対象を変更した方式（パターン2），及びXOR演算を行うパケットの数を減らした方式（パターン3,4）に対して，それぞれのパケット損失に対する耐性の検証・比較を行う．

#### 4.1.1 送信器の設定

既存研究[6][7]では，センサーから複数の送信器にパケットを送り，それぞれの送信器は送られてきたパケットを用いて符号化を行っていた．

例 7 (既存研究[7]より引用) センサーの数を9，冗長度を2，送信器の数を3とし，各送信器には表4.1に対応するパケットがセンサーから送られてくると仮定する．この場合， $R_1$ では $P_1, P_2, P_3, P_4, P_5, P_6$ ， $R_2$ では $P_4, P_5, P_6, P_7, P_8, P_9$ ， $R_3$ では $P_7, P_8, P_9, P_1, P_2, P_3$ をそれぞれ用いて符号化を行う．

表 4.1: 各送信器に届くパケット

送信器	センサーから送信器へ送られてくるパケット					
$R_1$	$P_1$	$P_2$	$P_3$	$P_4$	$P_5$	$P_6$
$R_2$	$P_4$	$P_5$	$P_6$	$P_7$	$P_8$	$P_9$
$R_3$	$P_7$	$P_8$	$P_9$	$P_1$	$P_2$	$P_3$

本研究では、簡単のためにセンサーからパケットが送られてくる送信器を1つにして、送信器がセンサーから送られてくる全てのパケットを用いて符号化出来るように設定する。

例 8 センサーの数を9, 冗長度を2, パケットが送られる送信器は $R_1$ のみとし, 送信器には表4.2に対応するパケットがセンサーから送られてくると仮定する. この場合,  $R_1$ では $P_1, P_2, \dots, P_9$ 全てのパケットを用いて符号化を行うことができる.

表 4.2: 送信器 $R_1$ に届くパケット

送信器	センサーから送信器へ送られてくるパケット								
$R_1$	$P_1$	$P_2$	$P_3$	$P_4$	$P_5$	$P_6$	$P_7$	$P_8$	$P_9$

以降, 送信器に届いたパケットの種類やXOR符号化方式を表すのに用いる表では, 送信器の項目は省略する.

#### 4.1.2 検証するXOR符号化方式の構成パターン

今回の検証では, 次の4つのXOR符号化方式のパターンについて検証する. さらに, これらの方式が符号化を行うことでどれほどパケット損失の耐性が高くなるのかを調査するため, どのパケットに対してもXOR演算を行っていない方式(符号化なし)についても検証する.

##### パターン1

既存研究[7]において, センサーの数を9, 冗長度を2とした時, 提案アルゴリズムを元に構成した方式におけるパケット損失の耐性評価が行われた. この中で, 最も耐性が高い結果を得た方式(表4.3)をパターン1とする. また, この方式をグラフ表記したものは図4.1となる. 以降は, このパターン1を元に構成したもの(パターン2)と, グラフ表記した際に形状が綺麗に整ったもの(パターン3,4)を考える.

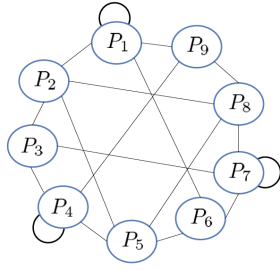


図 4.1: パターン1のグラフ

表 4.3: パターン1の方式

送信器からMSへ送られるパケット					
$P_1$	$P_2 \oplus P_5$	$P_3 \oplus P_4$	$P_4 \oplus P_5$	$P_5 \oplus P_6$	$P_6 \oplus P_1$
$P_4$	$P_5 \oplus P_8$	$P_6 \oplus P_7$	$P_7 \oplus P_8$	$P_8 \oplus P_9$	$P_9 \oplus P_4$
$P_7$	$P_8 \oplus P_2$	$P_9 \oplus P_1$	$P_1 \oplus P_2$	$P_2 \oplus P_3$	$P_3 \oplus P_7$

## パターン2

パターン2は、表4.4の方式とする。これは、パターン1の  $P_2 \oplus P_5, P_3 \oplus P_4$  の部分を、赤字のように  $P_2 \oplus P_4, P_3 \oplus P_5$  に変えたものである。また、この方式をグラフ表記したものは図4.2となる。

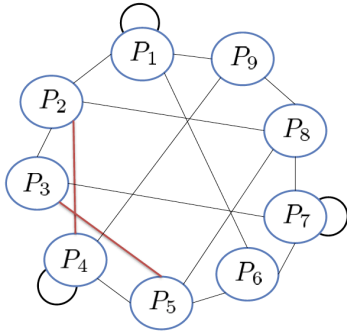


図 4.2: パターン2のグラフ

表 4.4: パターン2の方式

送信器からMSへ送られるパケット					
$P_1$	$P_2 \oplus P_4$	$P_3 \oplus P_5$	$P_4 \oplus P_5$	$P_5 \oplus P_6$	$P_6 \oplus P_1$
$P_4$	$P_5 \oplus P_8$	$P_6 \oplus P_7$	$P_7 \oplus P_8$	$P_8 \oplus P_9$	$P_9 \oplus P_4$
$P_7$	$P_8 \oplus P_2$	$P_9 \oplus P_1$	$P_1 \oplus P_2$	$P_2 \oplus P_3$	$P_3 \oplus P_7$

## パターン3

パターン3は、表4.5とする。これは、グラフ表記すると図4.3のように、3つの成分で構成される。それぞれの成分は、各頂点にループが付いている完全グラフ（どの2頂点を選んでも辺で繋がっているグラフ）である。

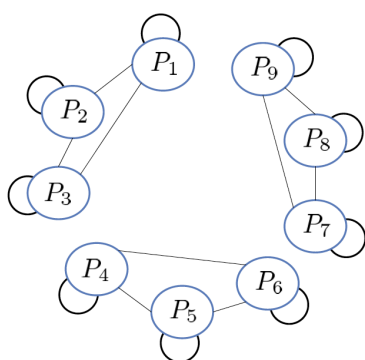


図 4.3: パターン 3 のグラフ

表 4.5: パターン 3 の方式

送信器から MS へ送られるパケット					
$P_1$	$P_2$	$P_3$	$P_4 \oplus P_5$	$P_5 \oplus P_6$	$P_6 \oplus P_4$
$P_4$	$P_5$	$P_6$	$P_7 \oplus P_8$	$P_8 \oplus P_9$	$P_9 \oplus P_7$
$P_7$	$P_8$	$P_9$	$P_1 \oplus P_2$	$P_2 \oplus P_3$	$P_3 \oplus P_1$

#### パターン 4

パターン 4 は，表 4.6 とする．これは，グラフ表記すると図 4.4 となる．パターン 4 のグラフでは， $P_1$  から  $P_9$  で長さ 9 のサイクルを構成し，またそれぞれの頂点にループが付いている．

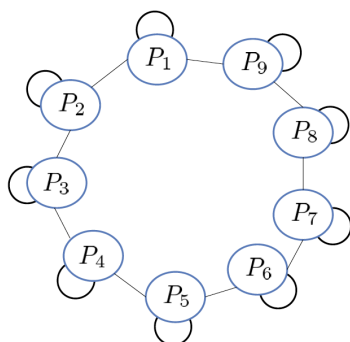


図 4.4: パターン 4 のグラフ

表 4.6: パターン 4 の方式

送信器から MS へ送られるパケット					
$P_1$	$P_2$	$P_3$	$P_4 \oplus P_5$	$P_5 \oplus P_6$	$P_6 \oplus P_7$
$P_4$	$P_5$	$P_6$	$P_7 \oplus P_8$	$P_8 \oplus P_9$	$P_9 \oplus P_1$
$P_7$	$P_8$	$P_9$	$P_1 \oplus P_2$	$P_2 \oplus P_3$	$P_3 \oplus P_4$

#### 符号化なし

符号化なしの方式は，表 4.7 とする．これは，グラフ表記すると図 4.5 となる．何も符号化をせず，送信器から MS へ元々のパケットが 2 回ずつ送信されている．

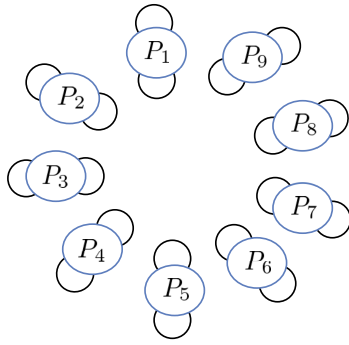


図 4.5: 符号化なしのグラフ

表 4.7: 符号化なしの方式

送信器からMSへ送られるパケット					
$P_1$	$P_2$	$P_3$	$P_4$	$P_5$	$P_6$
$P_4$	$P_5$	$P_6$	$P_7$	$P_8$	$P_9$
$P_7$	$P_8$	$P_9$	$P_1$	$P_2$	$P_3$

### 4.1.3 検証環境

それぞれのパターンについて，次の手順でパケット損失に対する耐性を検証した．

手順1：パターンに沿って符号化された18個のパケットを用意する（この手順が，送信器で符号化が行われた部分に該当する）．

手順2：それぞれのパケットを，確率 $q$ で消滅させる（この手順が，通信中にパケット損失が発生した部分に該当する）．

手順3：残ったパケットを用いて復号を行い， $P_1, P_2, \dots, P_9$ 全てのパケットが復号できれば成功とする（この手順が，MSで復号を行う部分に該当する）．

例 9 上記の手順を，パターン1を例に取って説明する．

#### 手順1

パターン1に沿って符号化された表4.8の18個のパケットを用意する．

表 4.8: 手順1時点でのパケット

用意されたパケット					
$P_1$	$P_2 \oplus P_5$	$P_3 \oplus P_4$	$P_4 \oplus P_5$	$P_5 \oplus P_6$	$P_6 \oplus P_1$
$P_4$	$P_5 \oplus P_8$	$P_6 \oplus P_7$	$P_7 \oplus P_8$	$P_8 \oplus P_9$	$P_9 \oplus P_4$
$P_7$	$P_8 \oplus P_2$	$P_9 \oplus P_1$	$P_1 \oplus P_2$	$P_2 \oplus P_3$	$P_3 \oplus P_7$

#### 手順2

それぞれのパケットを，確率 $q$ で消滅させる．今回は，表4.9のように $P_2 \oplus P_5, P_6 \oplus P_1, P_3 \oplus P_7, P_8 \oplus P_2, P_9 \oplus P_1$ が消滅したとする．

表 4.9: 確率  $q$  であるパケットが消滅した後

手順2時点で残っているパケット					
$P_1$		$P_3 \oplus P_4$	$P_4 \oplus P_5$	$P_5 \oplus P_6$	
$P_4$	$P_5 \oplus P_8$	$P_6 \oplus P_7$	$P_7 \oplus P_8$	$P_8 \oplus P_9$	$P_9 \oplus P_4$
$P_7$			$P_1 \oplus P_2$	$P_2 \oplus P_3$	

### 手順3

残ったパケットを用いて復号を行う。既に  $P_1, P_4, P_7$  はMSが受信できているので、残りのパケット  $P_2, P_3, P_5, P_6, P_8, P_9$  が復号できれば良い。

まず、 $P_2, P_5, P_8$  について調べる。これらのパケットは、手順2で消滅しなかった  $P_1, P_4, P_7$  と  $P_1 \oplus P_2, P_4 \oplus P_5, P_7 \oplus P_8$  を用いて

$$P_1 \oplus (P_1 \oplus P_2) = P_2$$

$$P_4 \oplus (P_4 \oplus P_5) = P_5$$

$$P_7 \oplus (P_7 \oplus P_8) = P_8$$

と復号できる。

次に、 $P_3, P_6, P_9$  について調べる。これらのパケットは、先ほど復号した  $P_2, P_5, P_8$  と手順2で消滅しなかった  $P_2 \oplus P_3, P_5 \oplus P_6, P_8 \oplus P_9$  を用いて

$$P_2 \oplus (P_2 \oplus P_3) = P_3$$

$$P_5 \oplus (P_5 \oplus P_6) = P_6$$

$$P_8 \oplus (P_8 \oplus P_9) = P_9$$

と復号できる。

以上より、 $P_2, P_3, P_5, P_6, P_8, P_9$  が復号された。さらに手順2で消滅しなかった  $P_1, P_4, P_7$  と合わせると、 $P_1, P_2, \dots, P_9$  全て存在するので通信成功とする。

この試行を複数回行い、試行の失敗回数（全てのパケットが復号できなかった回数）から次の復号誤り確率 (Decoding Error Probability, DEP) を定義して、それを評価の指標として用いた。

### 定義2

$$\text{復号誤り確率 (DEP)} = \frac{\text{失敗回数}}{\text{試行回数}}$$



なお，検証の試行回数は10万回， $q$ は0.1から0.5までを0.1刻みで変動させた．さらに，処理系にはMAGMAを用いた．

#### 4.1.4 検証結果

各パターンにおけるDEPは，図4.6のようになった．

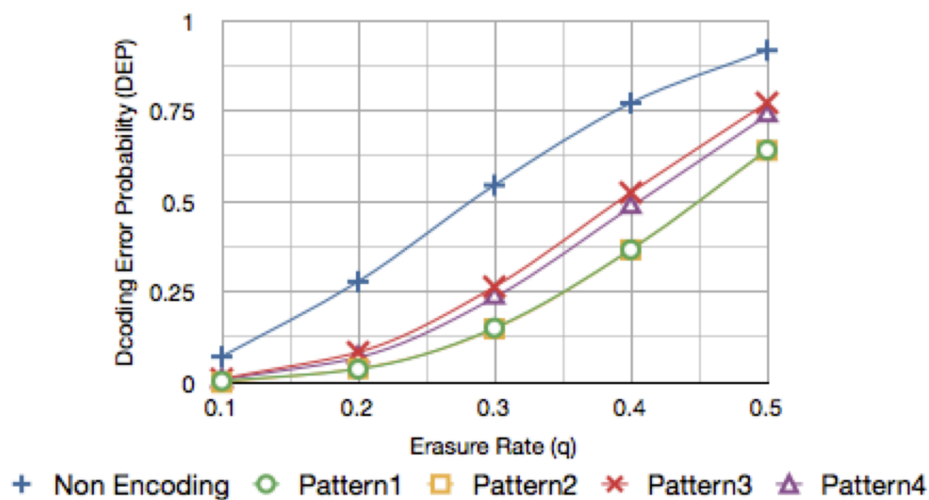


図 4.6: 各パターンにおけるDEP

#### 4.1.5 考察

まず，図4.6より，XOR符号化方式におけるパターン1,2,3,4に対して全てのDEPは，符号化しない場合に比べ低く抑えられた．評価した各 $q = 0.1, 0.2, 0.3, 0.4, 0.5$ において，特に $q = 0.3$ の時はパターン1と2のDEPは符号化しない場合よりも最大で0.3ほど低いことがわかる．つまり，これらの符号化方式を用いることで符号化を行わなかった場合よりパケット損失に対する耐性が高くなることがわかる．

さらに， $q$ が高くなると，パターン3と4のDEPはパターン1と2に比べて高くなる傾向にあることがわかる．また，評価した各 $q$ において，特に $q$ が0.3以上になると，DEPは最大で0.12ほど高くなることがわかる．また，パターン1とパターン2のDEPは $q$ の値によらず，ほぼ同値であった．パターン1と2は既存研究[7]のアルゴリズムに基づく方式なので，今回調

査した中では、このアルゴリズムを使ってXOR符号化方式を構成するパターンが最もパケット損失に対する耐性が高くなることがわかった。つまり、今回の検証では、既存研究[7]のアルゴリズムを元にした方式よりパケット損失に対する耐性が高くなる方式を見つけることができなかった。

## 4.2 XOR符号化方式の総当り検証

前節では、既存の方式の構成を変えたいくつかのパターンを考案し、それぞれの方式のパケット損失に対する耐性を検証した。しかし、符号化方式を構成する方法は他にも数多く存在するため、最もパケット損失に対する耐性が高い方式を発見するには、全ての方式を調べる必要がある。ところが、センサーの数や冗長度が大きくなると符号化方式の数が膨大になるため、全て調べるには非常に手間がかかる。そこで、本節では冗長度を2に固定し、センサーの数が少ない事例に絞ることによって符号化方式の数を少なくし、各方式のパケット損失に対する耐性を総当りで検証する。さらに、その中で最もパケット損失に対する耐性が高い符号化方式を見つけ出す。

### 4.2.1 検証準備

各符号化方式を総当りで検証するにあたり、第2章の定義1で定義されたグラフ  $G \in D(n, m)$  において復号可能な確率  $P_G$  を用いて比較する。前節で用いたDEPは、全てのパケットを復号できない確率であるので、あるグラフ  $G \in D(n, m)$  におけるDEPは確率  $P_G$  を用いると、

$$DEP = 1 - P_G \quad (4.1)$$

で求めることができる。また、あるグラフ  $G \in D(n, m)$  の  $P_G$  と、 $D(n, m)$  に属する全てのグラフ  $H$  の  $P_H$  を比較した時、

$$P_G \geq P_H \quad (4.2)$$

が全ての  $H$  で成り立てば、グラフ  $G$  は頂点数が  $n$ 、辺数が  $m$  のグラフの中で最もパケット損失に対する耐性が高いグラフである。

よって、式(4.2)が成り立つグラフ  $G \in D(n, m)$  を見つけることができれば、このグラフから構成するXOR符号化方式は式(4.1)よりセンサーの数

が $n$ ，冗長度が $\frac{m}{n}$ の方式の中で最もDEPが低くなる．なお，この検証では冗長度を2に固定するため，センサーの数，つまり頂点数を $n$ とすると辺数は $m = 2n$ となることに注意されたい．そこで，今回は式(4.2)が成り立つグラフ $G \in D(n, m)$ を見つけることで，最もDEPを低くするXOR符号化方式の発見を試みた．

なお，式(4.2)が成り立つグラフ $G \in D(n, m)$ を見つけるために，次の命題を用いた．

**命題 1**  $G$ を頂点数が $n$ ，辺数が $m$ の復号可能なグラフとし， $G$ から $x$ 本の辺を取り除いた際に復号可能となるグラフの個数を $C_x^G$ とする．今，任意のグラフ $H \in D(n, m)$ に対し，

$$C_x^G \geq C_x^H \quad (4.3)$$

が $x = 0, 1, \dots, m$ において成り立つならば， $G$ と $H$ の復号できる確率 $P_G$ と $P_H$ において

$$P_G \geq P_H \quad (4.4)$$

が成り立つ．つまり， $G$ は $D(n, m)$ で復号できる確率を最大にするグラフである．

### 証明

まず， $P_G$ は

$$P_G = \sum_{x=0}^m C_x^G q^{m-x} (1-q)^x$$

で定義されていることを復習しておく．次に，確率 $q$ は $0 \leq q \leq 1$ より，任意の $x = 0, 1, \dots, m$ について $q^{m-x} \geq 0$ ， $(1-q)^x \geq 0$ となるので

$$q^{m-x} (1-q)^x \geq 0 \quad (4.5)$$

が成り立つことに注意する．よって，全ての $x = 0, 1, \dots, m$ について式(4.3)が成り立てば

$$P_G - P_H \quad (4.6)$$

$$= \sum_{x=0}^m C_x^G q^{m-x} (1-q)^x - \sum_{x=0}^m C_x^H q^{m-x} (1-q)^x \quad (4.7)$$

$$= \sum_{x=0}^m (C_x^G - C_x^H) q^{m-x} (1-q)^x \geq 0 \quad (4.8)$$

が成り立つので,

$$P_G \geq P_H$$

を得る。(証明終)

各  $C_x^G$  を求めるには, 接続行列 (incidence matrix) を用いる. 接続行列とは, ある無向グラフの頂点を  $P_i$ , 辺を  $e_k$  とラベリングした時, 以下の条件で構成された  $|E| \times |V|$  の 0-1 行列である.

- 辺  $e_k$  が頂点  $P_i$  と  $P_j$  を繋いでいる時は行列成分を  $(k, i) = (k, j) = 1$  とする.
- 辺  $e_k$  が頂点  $P_i$  のループである時は行列成分を  $(k, i) = 1$  とする.
- 上の2つに該当しない行列成分は0とする.

例 10 図4.7にグラフの接続行列と, 対応する接続行列を示す. 例えば, 辺  $e_1$  は頂点  $P_1$  のループなので行列成分  $(1,1)$  が1になる. さらに, 辺  $e_3$  は頂点  $P_1$  と  $P_2$  を繋いでいるので行列成分  $(3,1)$  と  $(3,2)$  が1になる.

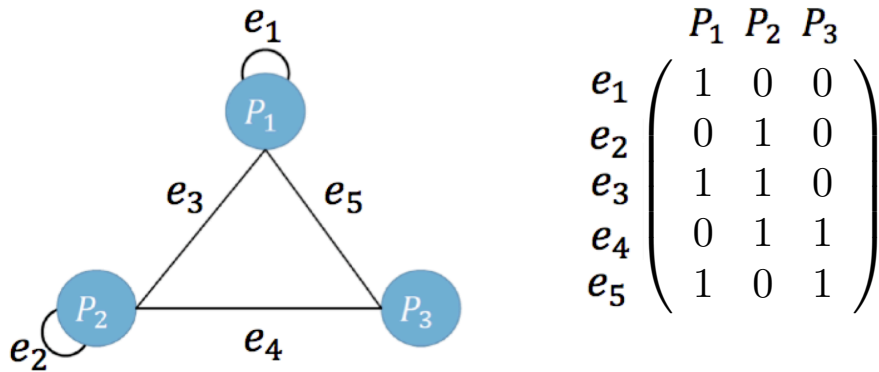


図 4.7: 接続行列の例

さらに, グラフ  $G$  から辺  $e_k$  が取り除かれた場合, グラフ  $G$  の接続行列の  $k$  番目の行成分を取り除く. また, 頂点数が  $n$  のグラフの接続行列の階数が  $F_2$  上で考えた時に  $n$  であれば,  $n$  個の一次独立な行ベクトルを用いて構成した行列から各  $P_i$  を一意に解くことが出来るため, このグラフは復号可能であることがわかる.

例 11 図 4.7 から  $e_1$  と  $e_4$  が取り除かれた場合，グラフと接続行列は図 4.8 となる．この接続行列の階数は 3 であるので，対応するグラフは復号可能である．一方，図 4.7 から  $e_1$  と  $e_2$  が取り除かれた場合，グラフと接続行列は図 4.9 となる．この接続行列の階数は 2 であるため，対応するグラフは復号可能でない．

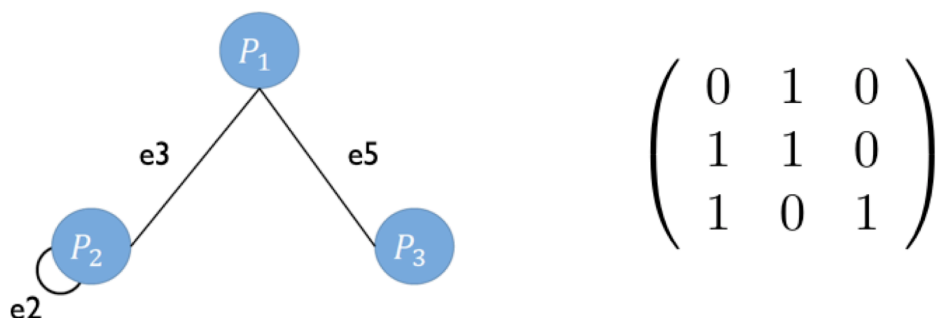


図 4.8: 復号可能なグラフの例

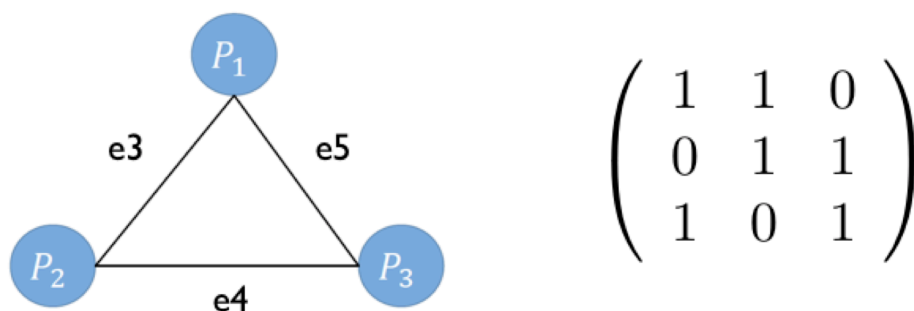


図 4.9: 復号可能でないグラフの例

つまり，頂点数が  $n$ ，辺数が  $m$  のグラフ  $G$  において， $G$  から  $x$  本辺を取り除いた際に復号可能となるグラフの個数は以下の手順で求められる．

手順 1: グラフ  $G$  の接続行列の行成分を  $x$  個取り除く ( $x$  個取り除くと， ${}_m C_x$  通りの接続行列ができる)．

手順 2: 各接続行列の階数を調べ，階数が  $n$  となる接続行列の個数が  $C_x^G$  となる．

そこで、センサーの数が $n$ で冗長度が2の場合（つまり、頂点数が $n$ で辺数が $2n$ で構成される全てのグラフ）に関して、接続行列を求めた。さらに、各接続行列から $x = 0, 1, 2, \dots, n$ における $C_x^G$ を求めて、全ての $x$ に対して式(4.2)が成り立つグラフ $G$ を見つけることを試みた。ただし、 $n$ が大きくなると検証対象となる接続行列は膨大になるため、今回は $n = 4, 5, 6$ の事例に対して検証を行った。また、処理系はMATLABを用いた。

#### 4.2.2 検証結果

各 $n$ におけるそれぞれの $x$ に対して、式4.3が成り立つグラフ $G$ 、及び各グラフ $G$ に対応するXOR符号化方式は以下のようになった。ただし、得られた各グラフ $G$ において同型なグラフが存在した場合、そのうち1つを示すことにする。同型なグラフとは、あるグラフの頂点を入れ替えると得られるグラフのことである。

例 12 図4.10の2つのグラフは同型である。何故なら、左のグラフの頂点 $P_2$ と $P_3$ を入れ替えると、右のグラフと同じ形になるからである。

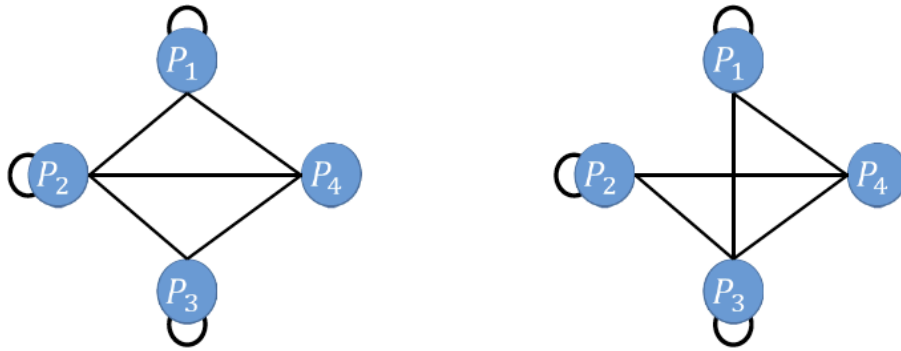


図 4.10: 同型なグラフの例

$n = 4$  の場合

各  $x = 0, 1, 2, 3, 4$  に対して式 4.3 が成り立つグラフ  $G$  は以下の図 4.11, 4.12, 各グラフに対応する XOR 符号化方式はそれぞれ表 4.10, 4.11 となった.

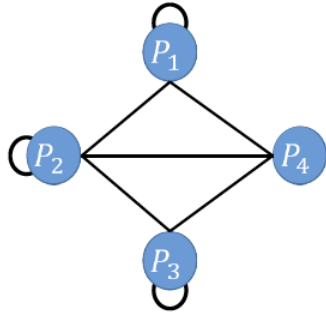


表 4.10: 図 4.11 に対応する方式

送信器から MS へ送られるパケット			
$P_1$	$P_3$	$P_2 \oplus P_3$	$P_3 \oplus P_4$
$P_2$	$P_1 \oplus P_2$	$P_2 \oplus P_4$	$P_4 \oplus P_1$

図 4.11: 各  $C_x^G$  が最大になるグラフ  $G$

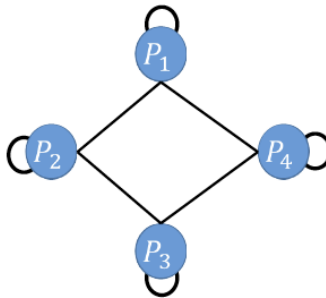


表 4.11: 図 4.12 に対応する方式

送信器から MS へ送られるパケット			
$P_1$	$P_3$	$P_1 \oplus P_2$	$P_2 \oplus P_3$
$P_2$	$P_4$	$P_3 \oplus P_4$	$P_4 \oplus P_1$

図 4.12: 各  $C_x^G$  が最大になるグラフ  $G$

グラフ  $G \in D(n, m)$  において,  $m \geq n$  の関係が成り立つ [7]. よって,  $x \geq m - n + 1$  の時はグラフ  $G$  から  $x$  本の辺を取り除くと復号可能ではなくなることがわかる. 故に,  $x \geq 5$  の時は  $C_x^G = 0$  となるので検証を行わなかった.

さらに, これらのグラフから得られる各  $C_x^G$  は表 4.12 のようになった.

表 4.12: 図 4.11, 4.12 のグラフと  $C_x^G$  の関係性

$x$	0	1	2	3	4
$C_x^G$	1	8	28	52	45

$n = 5$  の場合

各  $x = 0, 1, 2, 3, 4, 5$  に対して式 4.3 が成り立つグラフ  $G$  は以下の図 4.13, 4.14, 4.15, 各グラフに対応する XOR 符号化方式はそれぞれ表 4.13, 4.14, 4.15 となった.

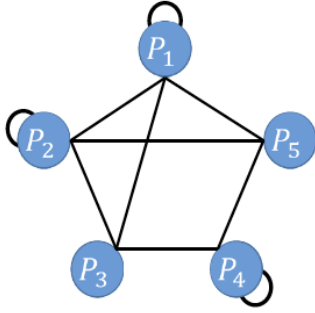


表 4.13: 図 4.13 に対応する方式

送信器から MS へ送られるパケット				
$P_1$	$P_4$	$P_2 \oplus P_3$	$P_3 \oplus P_4$	$P_5 \oplus P_1$
$P_2$	$P_1 \oplus P_2$	$P_2 \oplus P_5$	$P_4 \oplus P_5$	$P_1 \oplus P_3$

図 4.13: 各  $C_x^G$  が最大になるグラフ  $G$

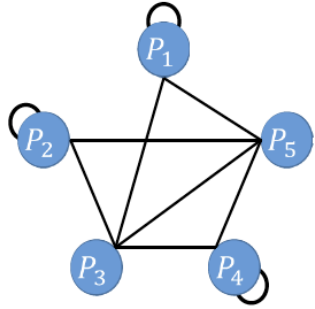


表 4.14: 図 4.14 に対応する方式

送信器から MS へ送られるパケット				
$P_1$	$P_4$	$P_2 \oplus P_3$	$P_3 \oplus P_4$	$P_5 \oplus P_1$
$P_2$	$P_2 \oplus P_5$	$P_3 \oplus P_5$	$P_4 \oplus P_5$	$P_1 \oplus P_3$

図 4.14: 各  $C_x^G$  が最大になるグラフ  $G$

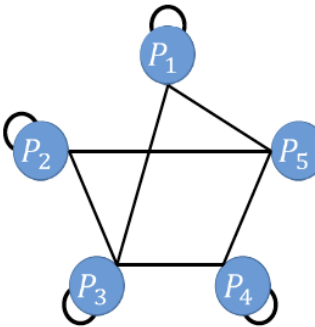


表 4.15: 図 4.15 に対応する方式

送信器から MS へ送られるパケット				
$P_1$	$P_3$	$P_1 \oplus P_3$	$P_3 \oplus P_4$	$P_5 \oplus P_1$
$P_2$	$P_4$	$P_2 \oplus P_3$	$P_4 \oplus P_5$	$P_5 \oplus P_2$

図 4.15: 各  $C_x^G$  が最大になるグラフ  $G$



また、これらのグラフから得られる各  $C_x^G$  は表 4.16 のようになった。

表 4.16: 図 4.13,4.14,4.15 のグラフと  $C_x^G$  の関係性

$x$	0	1	2	3	4	5
$C_x^G$	1	10	45	116	177	135

$n = 6$  の場合

各  $x = 0, 1, 2, 3, 4, 5, 6$  に対して式 4.3 が成り立つグラフ  $G$  は以下の図 4.16,4.17,4.18, 各グラフに対応する XOR 符号化方式はそれぞれ表 4.17,4.18,4.19 となった。

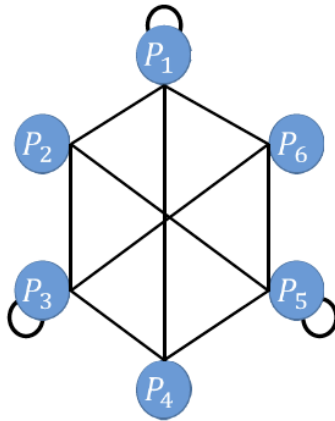


表 4.17: 図 4.16 に対応する方式

送信器から MS へ送られるパケット					
$P_1$	$P_5$	$P_2 \oplus P_3$	$P_3 \oplus P_4$	$P_5 \oplus P_6$	$P_6 \oplus P_1$
$P_3$	$P_1 \oplus P_2$	$P_2 \oplus P_5$	$P_4 \oplus P_5$	$P_1 \oplus P_4$	$P_3 \oplus P_6$

図 4.16: 各  $C_x^G$  が最大になるグラフ  $G$

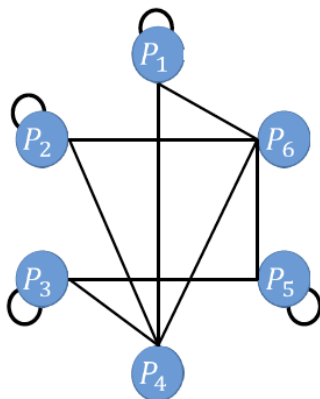


表 4.18: 図 4.17 に対応する方式

送信器から MS へ送られるパケット					
$P_1$	$P_5$	$P_3 \oplus P_5$	$P_3 \oplus P_4$	$P_5 \oplus P_6$	$P_6 \oplus P_1$
$P_3$	$P_4$	$P_2 \oplus P_4$	$P_2 \oplus P_6$	$P_4 \oplus P_6$	$P_1 \oplus P_4$

図 4.17: 各  $C_x^G$  が最大になるグラフ  $G$

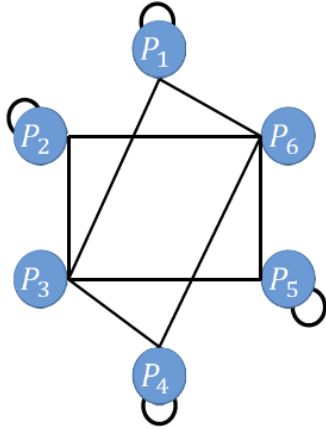


表 4.19: 図 4.18 に対応する方式

送信器から MS へ送られるパケット					
$P_1$	$P_5$	$P_1 \oplus P_3$	$P_3 \oplus P_4$	$P_5 \oplus P_6$	$P_6 \oplus P_1$
$P_2$	$P_4$	$P_2 \oplus P_3$	$P_2 \oplus P_6$	$P_3 \oplus P_5$	$P_4 \oplus P_6$

図 4.18: 各  $C_x^G$  が最大になるグラフ  $G$

また，これらのグラフから得られる各  $C_x^G$  は表 4.20 のようになった．

表 4.20: 図 4.16, 4.17, 4.18 のグラフと  $C_x^G$  の関係性

$x$	0	1	2	3	4	5	6
$C_x^G$	1	12	66	216	456	612	432

ここで，各  $n$  で得た  $C_x^G$  を最大にするグラフ  $G$  から得られる符号化方式と，既存研究 [7] の考案アルゴリズムから構成される符号化方式が一致するかを検証した．その結果， $n = 4$  でループの数が 4 のグラフ（図 4.12），及び  $n = 6$  でループの数が 3 のグラフ（図 4.16）は既存研究 [7] の考案アルゴリズムから構成される符号化方式と一致したが，他のグラフでは一致しなかった．よって，既存研究 [7] の考案アルゴリズムから構成される符号化方式は，必ずしもパケット損失に対する耐性が最も高くなる方式を導くとはいえないことがわかった．

### 4.2.3 考察

それぞれの  $n$  で得た  $C_x^G$  を最大にするグラフ  $G$  の共通点は，「グラフに含まれるループの数が 3 または 4」ということである．つまり，これらの結果を XOR 符号化方式に置き換えると，「センサーの数が  $n = 4, 5, 6$  かつ冗長度が 2 の時，XOR 演算しないパケットの個数は 3 または 4 の時にパケッ

ト損失に対する耐性が最も高くなる方式が存在する」ということである。これは、既存研究 [7] の結果において「冗長度が  $r$  の時、ループの数は  $2r - 1$  もしくは  $2r$  にしたほうが良い」と書かれていることと一致する。 $n$  の数が 7 以上になる場合においては、計算機の性能の関係で検証できなかったが、同様に XOR 演算しないパケットの個数が 3 または 4 の時に耐性が最も高くなるような方式が存在すると予想される。今後の課題として、 $n = 4, 5, 6$  の時の結果を用いて誤り耐性を最も高くするような方式のより詳細な理論的解析を行いたい。

## 第5章 提案手法における誤り耐性

本章では，提案する接続符号化について，パケット損失とビット反転誤りが両方同時に発生した場合に対する耐性の検証を行う．

### 5.1 接続符号化の性能検証実験

ここでは，XOR符号化とBCH(63,51)符号化を接続させることによって，パケット損失とBFEの両方に対する耐性がどれほど高くなるかをシミュレーションを用いて検証した．

#### 5.1.1 検証準備

この検証では，簡単のために発生する誤りの種類や送るパケットの数は次の条件とした．

- 通信中に発生する誤りは，パケット損失及びBFEとする．また，それぞれの誤りが発生する確率を $q$ 及び $p$ と置く．
- 9つの51-bitのパケットを用意し，冗長度を2としてMSへ送る．つまり，MSへ送るパケットの総数は18である．

なお，この検証で用いる処理系はMAGMAとした．

#### 5.1.2 検証する伝送システム

今回は，次の3つのタイプの符号化についてパケット損失とBFEに対する耐性を調査した．

タイプ1: BCH(63,51) 符号化のみを用いる符号化

タイプ2: BCH(63,51) 符号化を行った後に 63-bit XOR 符号化を行う符号化  
(図 2.13 に対応)

タイプ3: 51-bit XOR 符号化を行った後に BCH(63,51) 符号化を行う符号化  
(図 2.14 に対応)

また、各タイプを調査するために構築した伝送システムは、それぞれ次の通りである。

### タイプ1

構築した伝送システムは、図 5.1 の通りである。

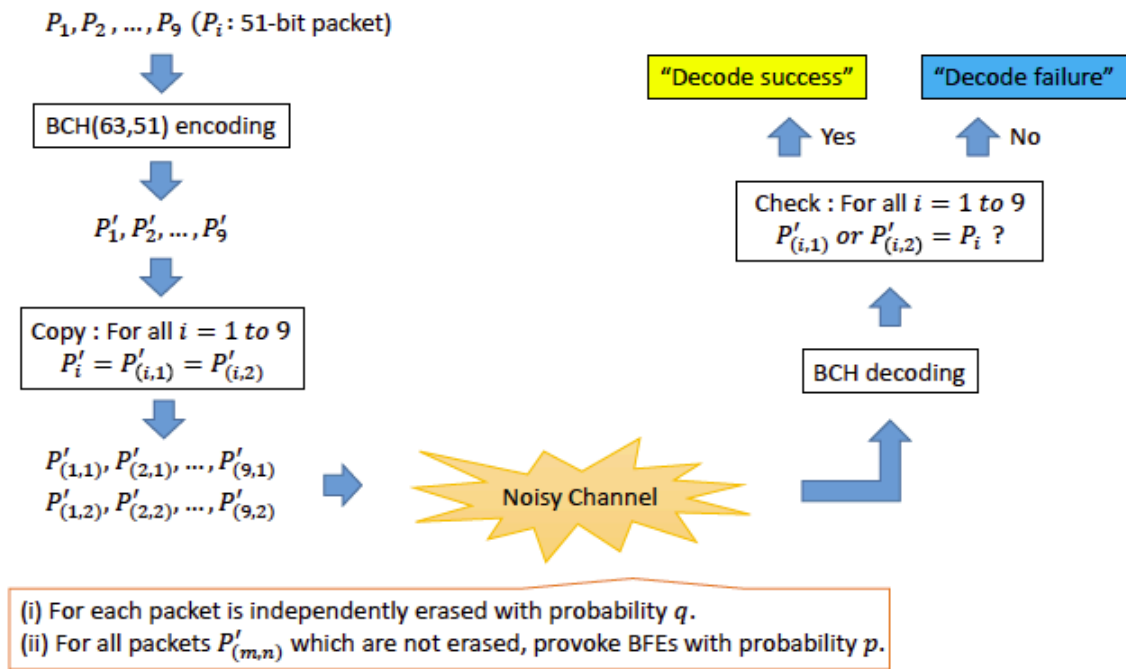


図 5.1: タイプ1の伝送システム

このシステムを用いたシミュレーション手順は、次の通りである。

手順1: 9つの 51-bit のパケット  $P_1, P_2, \dots, P_9$  をランダムに生成し、それぞれを BCH(63,51) 符号化を行う符号器で符号化する。符号化済みのパケットは  $P'_1, P'_2, \dots, P'_9$  と表すことにする。

手順2: 各  $P'_i$  ( $i = 1, 2, \dots, 9$ ) を  $P'_i = P'_{(i,1)} = P'_{(i,2)}$ , つまり  $P'_{(i,1)}$  と  $P'_{(i,2)}$  にコピー

する．この手順を行った後，パケットは全部で  $P'_{(1,1)}, P'_{(1,2)}, \dots, P'_{(9,1)}, P'_{(9,1)}$  の18個存在する．

手順3：各パケットに対して，それぞれ確率  $q$  でパケット自身を消滅させる．また，消滅せずに残った全てのパケット  $P'_{(i,j)} (j = 1, 2)$  に対して，それぞれのパケット内のビットに確率  $p$  でBFEを発生させる（この手順は，通信中に誤りが発生する場面を表している）．

手順4：消滅しなかった全てのパケット  $P'_{(i,j)}$  を用いて，復号を行う．もし， $P'_{(i,1)}$  か  $P'_{(i,2)}$  のうち最低でもどちらか一方が  $P_i$  のデータと一致したら， $P_i$  は復号出来たと判断する．これを全ての  $i$  について調べ，全ての  $P_i$  が復号出来た場合は「復号成功 (Decode success)」，出来なかった場合は「復号失敗 (Decode failure)」とみなす．

## タイプ2

構築した伝送システムは，図5.2の通りである．

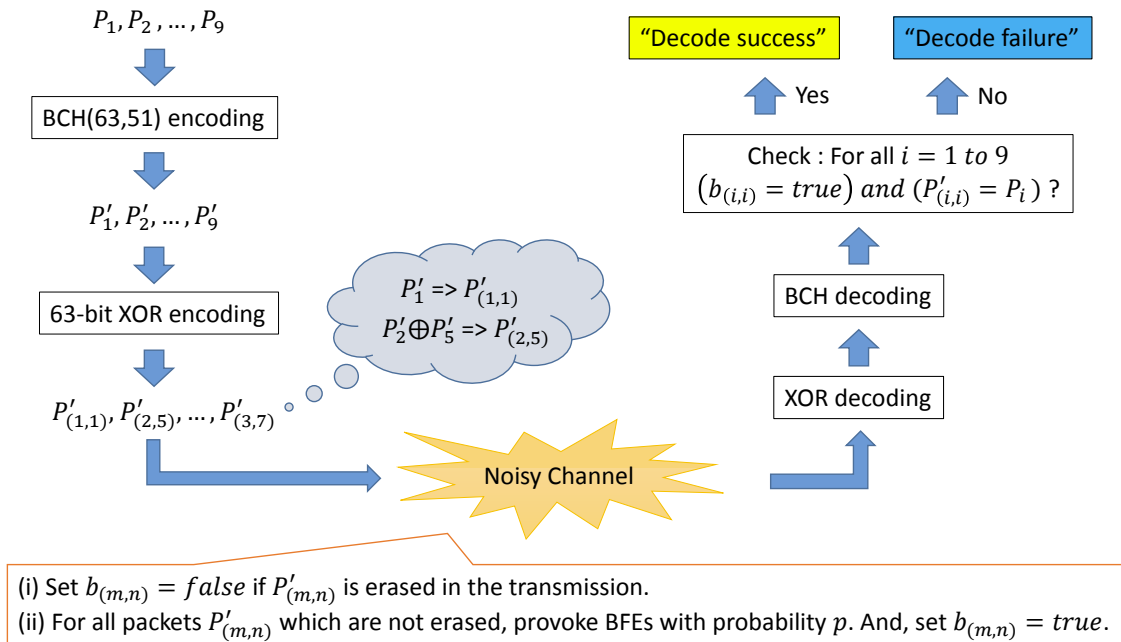


図 5.2: タイプ2の伝送システム

このシステムを用いたシミュレーション手順は，次の通りである．

手順1：タイプ1の手順1と同様に，9つの51-bit パケット  $P_i (i = 1, 2, \dots, 9)$  をランダムに生成し，及びそれぞれを BCH(63,51) 符号化を行う符号器で符号化する．

手順2：図5.3で示された XOR 符号化方式に従って，各  $P'_i$  から18個のパケットを生成する．ここでは，63-bit 同士のパケットの XOR 演算，つまり 63-bit XOR 符号化を15回行い， $P'_1, P'_4, P'_7$  のパケットは手順1で用意されたものをそのまま用いる．図5.3の方式は，第4章における検証でセンサーの数を9，冗長度を2とした時にパケット損失に対する耐性が高かったものである．

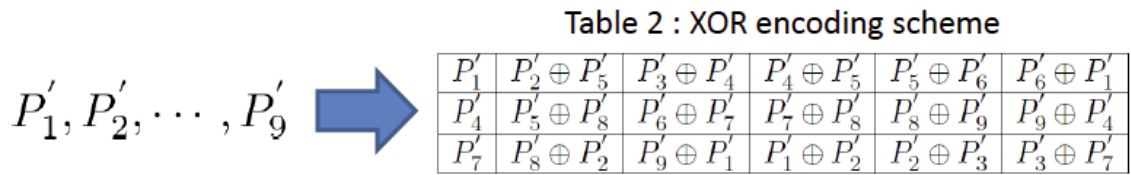


図 5.3: 各  $P'_i$  から XOR 符号化を行う

手順3：手順2で作成された18個のパケットは次のように表すことにする．

$$P'_{(m,n)} = \begin{cases} P'_m \oplus P'_n & (\text{when } m \neq n) \\ P'_m & (\text{when } m = n) \end{cases}$$

例えば， $P'_{(2,5)} = P'_2 \oplus P'_5$  及び  $P'_{(1,1)} = P'_1$  となる．

手順4： $b_{(m,n)} (1 \leq m, n \leq 9)$  を *true* または *false* の値をとるブール型の変数とする．この変数は，各  $b_{(m,n)}$  に対応する  $P'_{(m,n)}$  が存在している時， $b_{(m,n)} = \text{true}$  をとり，存在していない時は  $b_{(m,n)} = \text{false}$  をとる．手順4の段階では，手順2で作成された18個のパケットに対応する  $b_{(m,n)}$  は全て *true* であるが，それ以外は *false* である．例えば， $P'_{(2,5)} (= P'_2 \oplus P'_5)$  は手順4の段階で存在しているので， $b_{(2,5)} = \text{true}$  となるが， $P'_{(2,4)} (= P'_2 \oplus P'_4)$  は手順2で作成された18個のパケットに含まれていないので， $b_{(2,4)} = \text{false}$  となる．

手順5：手順2で作成された18個のパケットに対して，それぞれ確率  $q$  でパケットを消滅させる．消滅したパケットに対応する  $b_{(m,n)}$  は *false* に設定する（ここでは，通信中にパケット損失が発生する場面を表している）．また，消滅せずに残った（つまり，この時点で  $b_{(m,n)} = \text{true}$  である）全てのパケット  $P'_{(m,n)}$  に対して，それぞれのパケット内のビットに確率  $p$  で BFE を

発生させる（ここでは，通信中にビット反転誤りが発生する場面を表している）．

手順6：下記 Algorithm 1 に従って XOR 符号の復号を行う．

---

**Algorithm 1** XOR 符号の復号

---

**Require:**  $P'_{(m,n)}, b_{(m,n)} (m, n = 1, 2, \dots, 9)$

**Ensure:**  $P'_{(m,n)}, b_{(m,n)}$

```

1:  $c \leftarrow 0$ 
2: while  $c = 0$  do
3:    $c \leftarrow 1$ 
4:   for  $m = 1$  to 9 do
5:     if  $b_{(m,m)}$  is true then
6:       for  $n = 1$  to 9 do
7:         if  $(b_{(m,n)}$  is true) and  $(b_{(n,n)}$  is false) then
8:            $P'_{(n,n)} \leftarrow P'_{(m,m)} \oplus P'_{(m,n)}$ 
9:            $b_{(n,n)} \leftarrow \text{true}$ 
10:         $c \leftarrow 0$ 
11:       end if
12:     end for
13:   end if
14: end for
15: end while

```

---

このアルゴリズムは，ある  $P'_{(m,m)}$  が存在する（つまり， $b_{(m,m)} = \text{true}$ ）時，この  $P'_{(m,m)}$  を用いて復号が行える  $P'_{(m,n)}$ （つまり， $b_{(m,n)} = \text{true}$ ）を探し出して  $P'_{(n,n)}$  の復号を行う．ただし，一度復号した  $P'_{(n,n)}$  は，他の  $P'_{(o,o)}$  ( $o = 1, 2, \dots, 9$ ) ( $o \neq m$ ) と  $P'_{(o,n)}$  が存在しても，新しく  $P'_{(n,n)}$  を復号しない．さらに，ある  $P'_{(m,m)}$  が存在するかどうか  $P'_{(1,1)}, P'_{(2,2)}, \dots, P'_{(9,9)}$  を順番に調べるので，このアルゴリズムを用いると復号できるパケットが揃うまでの復号手順が1通りに定まり，かつ最短で求めることができる．

手順7：手順6まで得られたパケットに対して，BCH(63,51)復号を行う．この復号が終わった段階で，各  $i$  において  $P'_{(i,i)}$  と  $P_i$  のデータが一致したら， $P_i$  は復号出来たと判断する．これを全ての  $i$  について調べ，全ての  $P_i$  が復号出来た場合は「復号成功 (Decode success)」，出来なかった場合は「復号失敗 (Decode failure)」とみなす．



### タイプ3

構築した伝送システムは，図5.4の通りである．

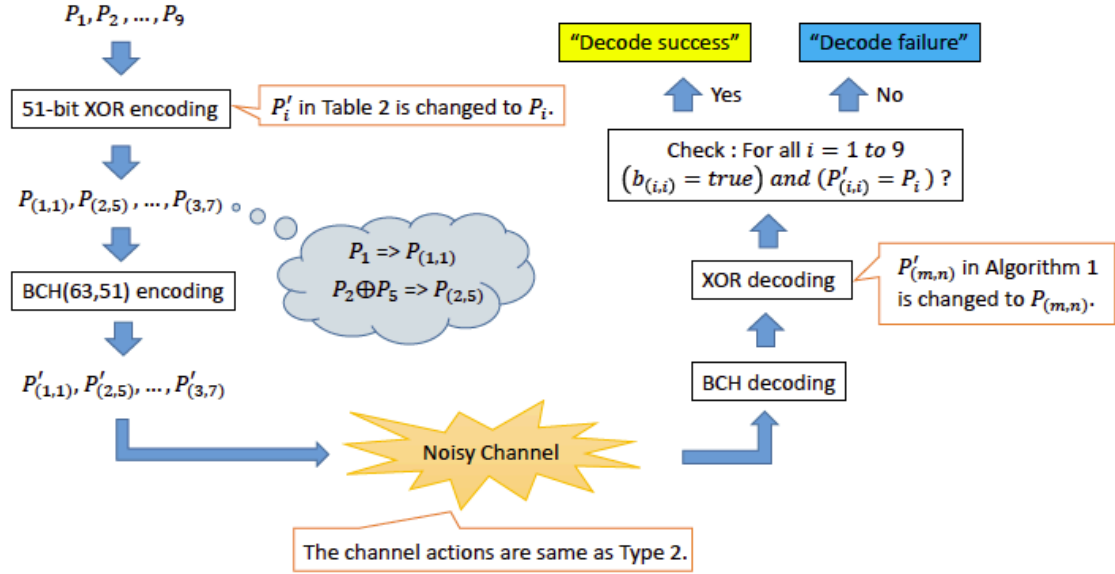


図 5.4: タイプ3の伝送システム

このシステムを用いたシミュレーション手順は，タイプ2の手順においてXOR符号化とBCH(63,51)符号化の順序を逆にしたものである．つまり，タイプ2の以下の点を変更したものとなる．

- 手順1のBCH(63,51)符号化を51-bit XOR符号化に変更する．
  - つまり，図5.3の各 $P'_i$ を $P_i$ に変更したものを用いる．
- 63-bit XOR符号化をBCH(63,51)符号化に変更する．
- Algorithm 1における各 $P'_{(m,n)}$ を $P_{(m,n)}$ に変更する．
  - ただし， $m \neq n$ の時は $P_{(m,n)} = P_m \oplus P_n$ ， $m = n$ の時は $P_{(m,m)} = P_m$ となる．

各タイプにおいて試行を複数回行い，復号失敗の回数から第4章の定義2で定義したDEPを求め，それを評価の指標に用いた．また，検証の試行回数は10万回， $q$ は0から0.5までを0.1刻みで変動させた．さらに， $p$ は0.001から0.006までを0.001刻みで変動させた．

### 5.1.3 連接符号化の性能検証結果

各  $q$  における,  $p$  と DEP の関係は図 5.5 から図 5.10 までとなった.

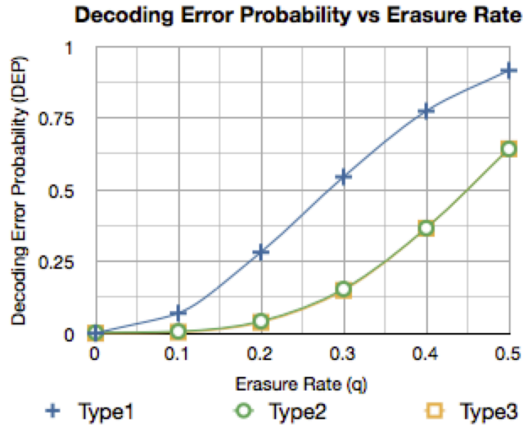


図 5.5:  $p = 0.001$

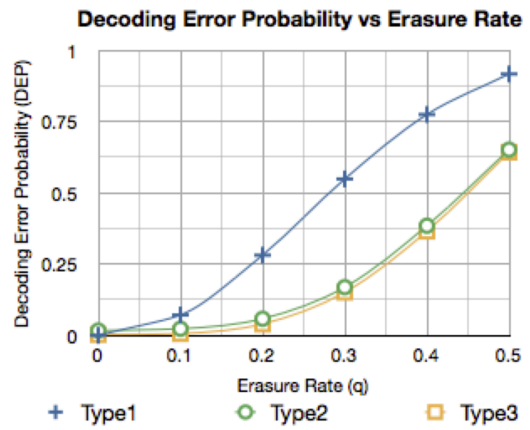


図 5.6:  $p = 0.002$

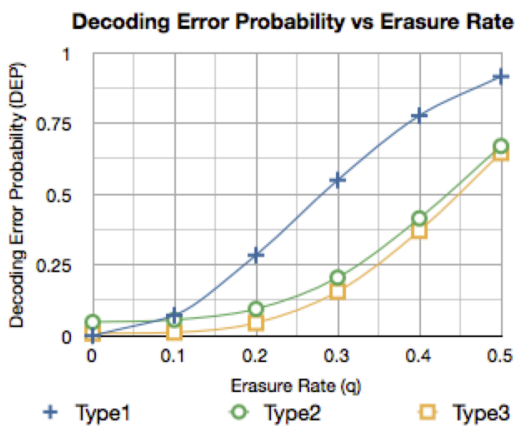


図 5.7:  $p = 0.003$

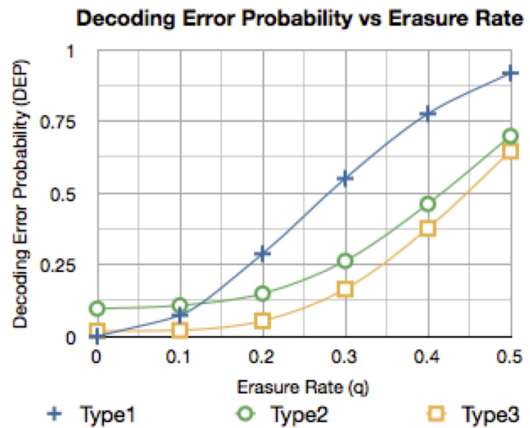


図 5.8:  $p = 0.004$

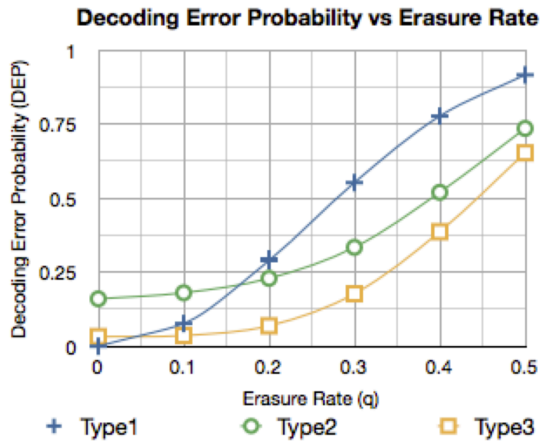


図 5.9:  $p = 0.005$

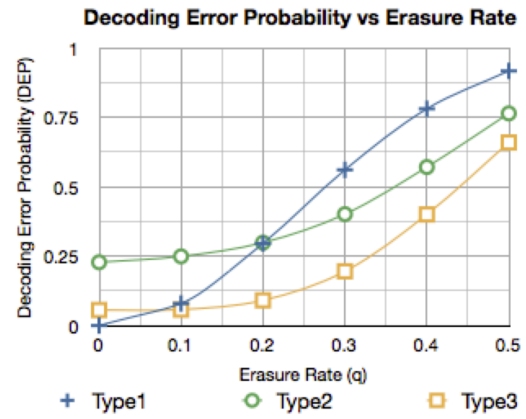


図 5.10:  $p = 0.006$

#### 5.1.4 考察

図 5.5 から図 5.10 の結果より、次のことが考察される。

考察 1: パケット損失が発生する確率  $q$  が高い時は ( $0.2 \leq q \leq 0.5$  の範囲では), 常にタイプ 2 とタイプ 3 の DEP はタイプ 1 の DEP より低くなる。

考察 2: BFE が発生する確率  $p$  が高い時は (確率が 0.003 以上になると), 常にタイプ 2 の DEP はタイプ 3 の DEP より高くなる。

考察 3:  $q$  が低くかつ  $p$  が高い時 (例えば,  $q = 0, p = 0.006$  の時), タイプ 1 の DEP はタイプ 2 とタイプ 3 の DEP より高くなる。

考察 1 より, BCH 符号化単体で符号化を行うよりも XOR 符号化と組み合わせる符号化の方が, 通信中にパケット損失が発生する確率が上がるほど有効であることがわかった。

考察 2 より, BFE が発生する確率が高くなればなるほど, XOR 符号化の後に BCH 符号化を行う方が有効であることがわかった。これは, 復号の際に BCH 符号の復号より XOR 符号の復号の方が先に行われていたからだと考えられる。XOR 演算は, 前節で述べたように復号によって BFE を増やしてしまう恐れがある。つまり, タイプ 2 では復号の際に先に XOR 符号の復号を行ったことによって, 手順 5 で発生した BFE がさらに増えてしまう可能性がある。そのため, 後の BCH 符号の復号でも修正しきれないほどの BFE が増えてしまったため, DEP が上がってしまったと考えられる。

考察3より、BFEが発生する確率が高く、かつパケット損失が発生する確率が低くなると、BCH符号化単体の方がXOR符号化と組み合わせるより有効であることがわかった。今回の検証では、タイプ1は $P'_{(i,1)}$ と $P'_{(i,2)}$ の両方が「復号が終わった段階で存在しない、またはデータが $P_i$ と一致しない」状態を復号失敗とした。つまり、タイプ1で復号失敗となるには、ある $P_i$ において $P'_{(i,1)}$ と $P'_{(i,2)}$ のどちらも受信されない、もしくは $P'_{(i,1)}$ か $P'_{(i,2)}$ のどちらかのパケットが存在していても、 $P_i \neq P'_{(i,1)}$ かつ $P_i \neq P'_{(i,2)}$ が成立していなければならない。一方、他のタイプでは「復号が終わった段階で $P'_{(i,i)}$ が存在しない、または $P_i$ と一致しない」状態を復号失敗とした。もし、復号が終わった段階である $i$ における $P'_{(i,i)}$ にBFEが含まれていた場合は、すぐに復号失敗となる。パケット損失が発生する確率が低い環境であれば受信パケットの数が多くなるが、同時にBFEが発生する確率が高い環境だと受信パケットにBFEを含む確率も高くなる。つまり、この場合はBCH符号の復号を行ってもパケットのBFEを修正できる確率も低くなる。よって、復号が終わった段階で存在するパケットから復号失敗の原因となるパケットを取り除ける場合があるタイプ1の方が、他のタイプと比べてDEPが低くなると考えられる。

以上の考察より、 $p$ や $q$ の値によって有効な符号化は変わってくると考えられる。

## 5.2 パケット数を少なくした場合の検証

前節では用意するパケット数を9個に設定したが、これより少ないパケットの数にした場合も、同様の結果が出るのかを調査した。前章では、冗長度を2に固定した際、センサーの数が4個、5個、6個の場合にパケット損失に対する耐性が最も高いXOR符号化方式を求めた。そこで、これらの方式を用いて、用意するパケットの数が4個、5個、6個の時のパケット損失とBFEに対する耐性を調査した。

### 5.2.1 用意するパケット数が4個の時のDEP

各タイプのシミュレーション手順の流れは、基本的に同じである。しかし、用意するパケットの数を9個( $P_1, P_2, \dots, P_9$ )から4個( $P_1, P_2, P_3, P_4$ )に変更した所に注意されたい。また、冗長度は2なので、MSへ送られるパケットの総数

は8になる．つまり，各タイプで用いていた $i$ は $i = 1, 2, \dots, 9$ から $i = 1, 2, 3, 4$ に変更する．また，タイプ2及び3で用いた $m, n$ は $m, n = 1, 2, \dots, 9$ から $m, n = 1, 2, 3, 4$ に変更する．

さらに，タイプ2及び3で用いたXOR符号化方式は，前章で求めた表5.1を用いた．

表 5.1: パケット数が4個の時に用いる方式

$P_1$	$P_3$	$P_2 \oplus P_3$	$P_3 \oplus P_4$
$P_2$	$P_1 \oplus P_2$	$P_2 \oplus P_4$	$P_4 \oplus P_1$

以上を加味すると，各 $q$ における $p$ とDEPの関係は図5.11から図5.16までとなった．

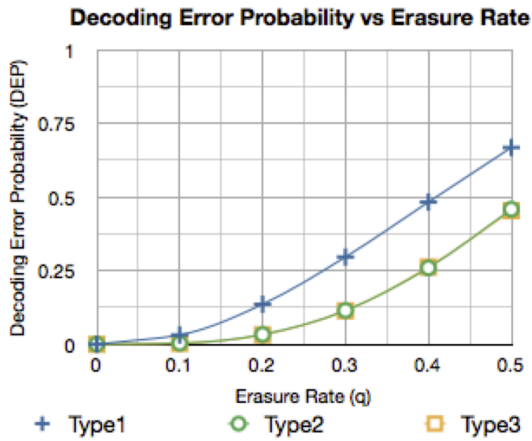


図 5.11:  $p = 0.001$

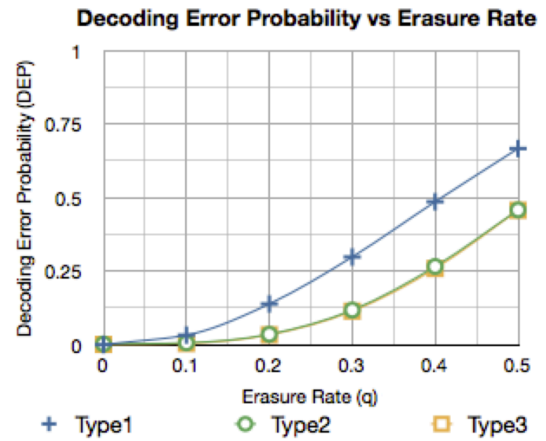


図 5.12:  $p = 0.002$

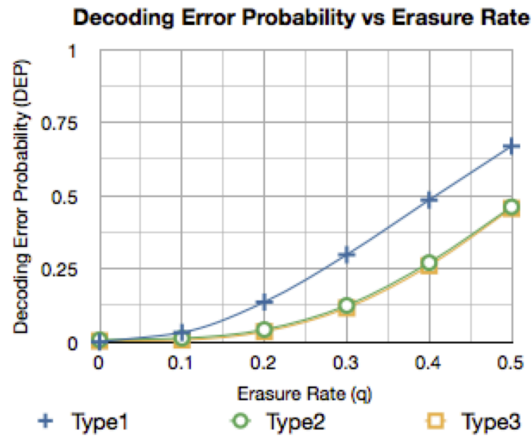


図 5.13:  $p = 0.003$

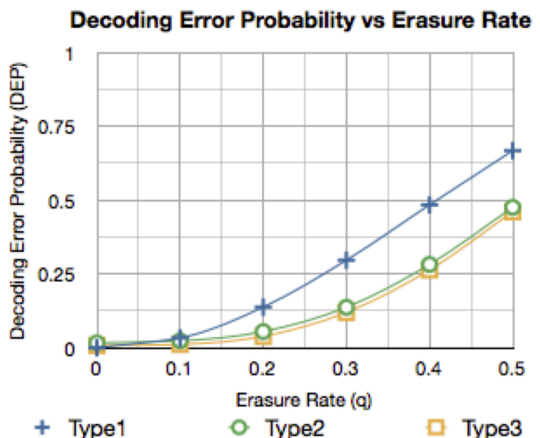


図 5.14:  $p = 0.004$

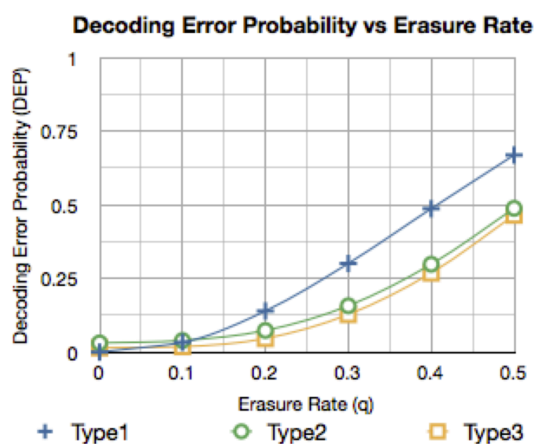


図 5.15:  $p = 0.005$

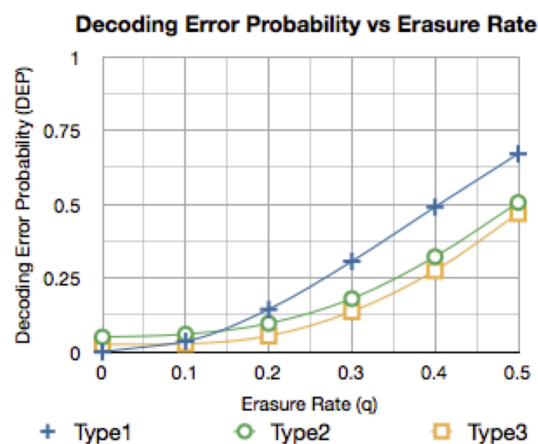


図 5.16:  $p = 0.006$

## 5.2.2 用意するパケット数が5個の時のDEP

パケットの数が4個の時と同様にして、パケット数が5個の時のDEPについても考える。タイプ2及び3で用いたXOR符号化方式は、前章で求めた表5.2を用いた。

表 5.2: パケット数が5個の時に用いる方式

$P_1$	$P_4$	$P_2 \oplus P_3$	$P_3 \oplus P_4$	$P_5 \oplus P_1$
$P_2$	$P_1 \oplus P_2$	$P_2 \oplus P_5$	$P_4 \oplus P_5$	$P_1 \oplus P_3$

以上を加味すると、各 $q$ における $p$ とDEPの関係は図5.17から図5.22までとなった。

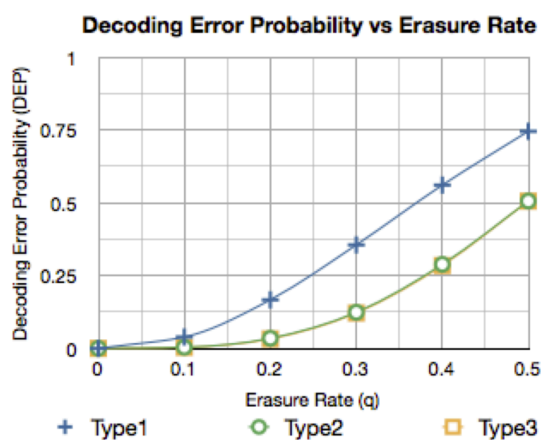


図 5.17:  $p = 0.001$

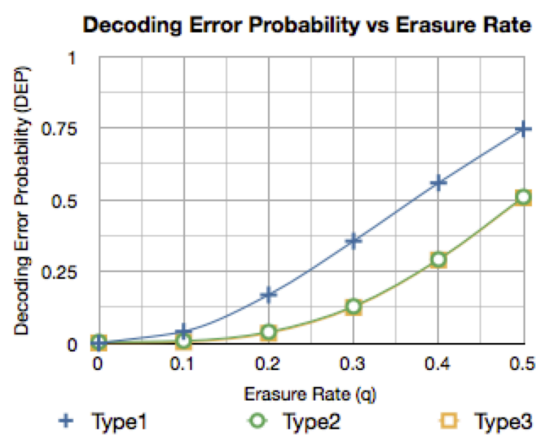


図 5.18:  $p = 0.002$

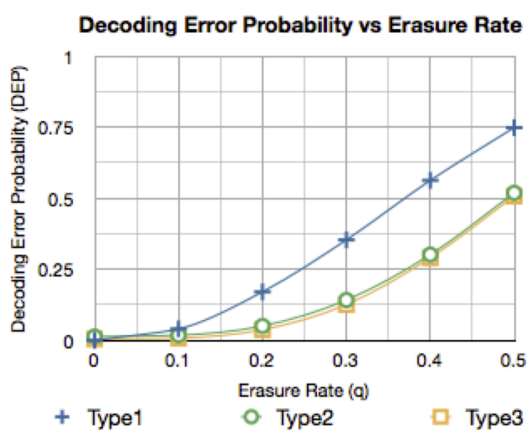


図 5.19:  $p = 0.003$

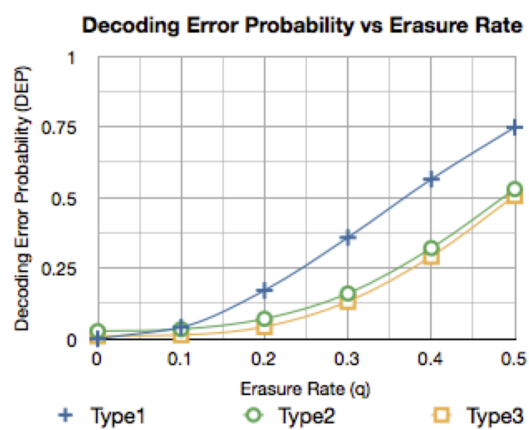


図 5.20:  $p = 0.004$

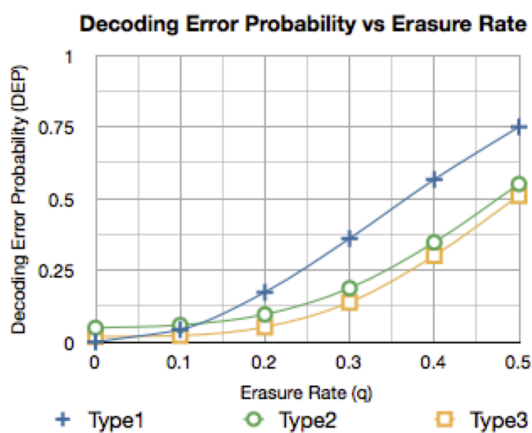


図 5.21:  $p = 0.005$

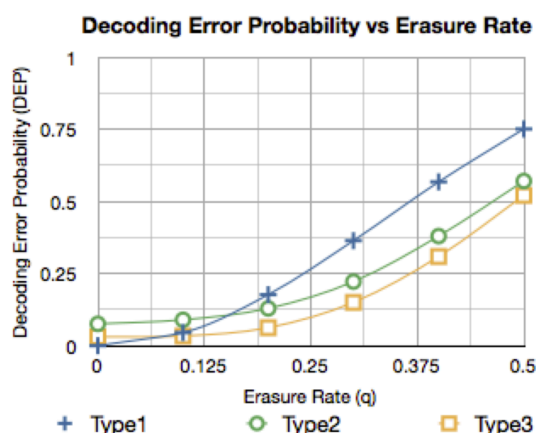


図 5.22:  $p = 0.006$

### 5.2.3 用意するパケット数が6個の時のDEP

最後に、パケットの数が6個の時のDEPについても調査した。タイプ2及び3で用いたXOR符号化方式は、前章で求めた表5.3を用いた。

表 5.3: パケット数が6個の時に用いる方式

$P_1$	$P_5$	$P_2 \oplus P_3$	$P_3 \oplus P_4$	$P_5 \oplus P_6$	$P_6 \oplus P_1$
$P_3$	$P_1 \oplus P_2$	$P_2 \oplus P_5$	$P_4 \oplus P_5$	$P_1 \oplus P_4$	$P_3 \oplus P_6$

以上を加味すると、各 $q$ における $p$ とDEPの関係は図5.23から図5.28までとなった。

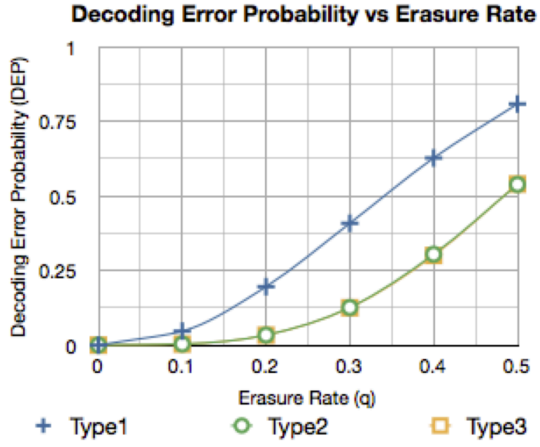


Figure 5.23:  $p = 0.001$

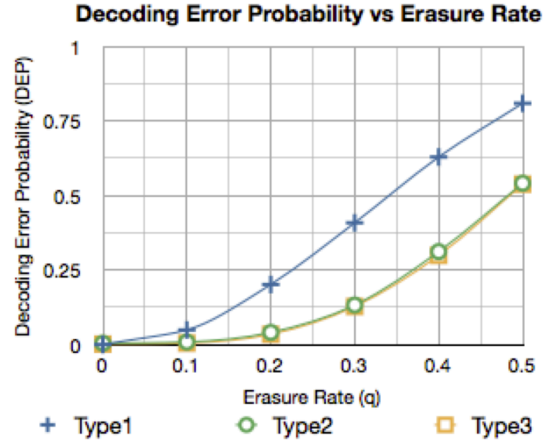


Figure 5.24:  $p = 0.002$

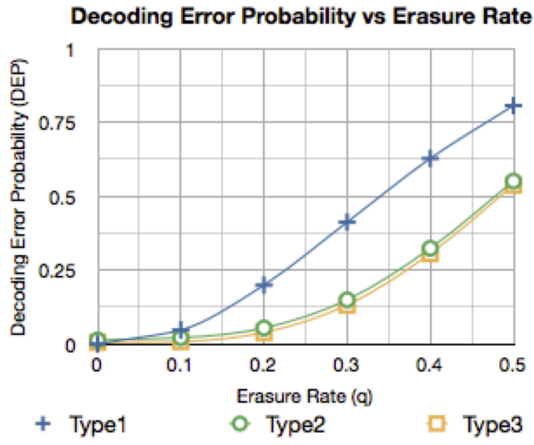


Figure 5.25:  $p = 0.003$

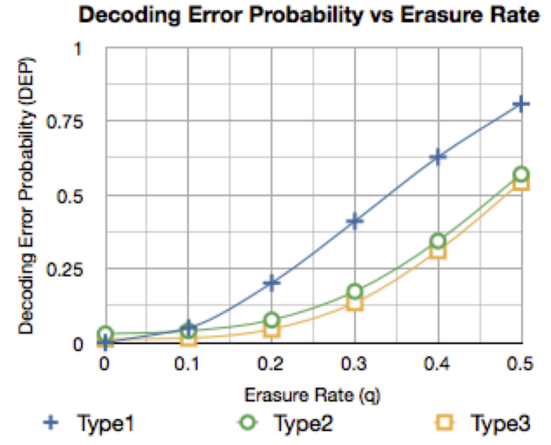


Figure 5.26:  $p = 0.004$

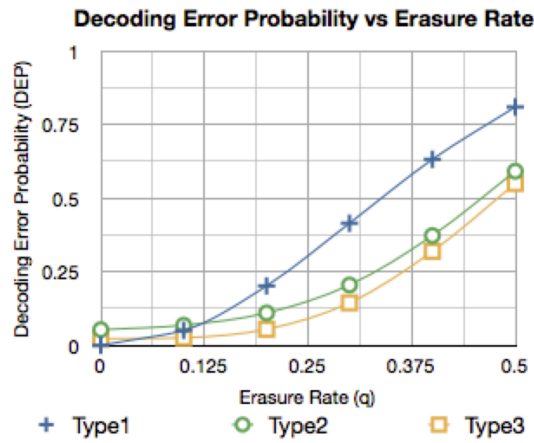


Figure 5.27:  $p = 0.005$

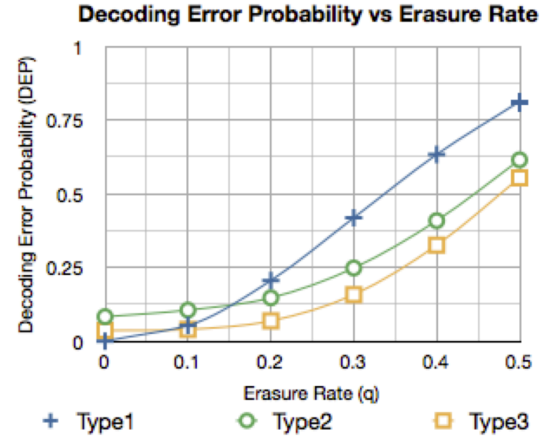


Figure 5.28:  $p = 0.006$



#### 5.2.4 考察

用意するパケットの数が4個、5個、6個の時の各 $p$ における $q$ とDEPの関係は、用意するパケットの数が9個の時のものと比較してみると、前節で述べた3つの現象と同様の傾向が見られることがわかった。つまり、用意するパケットの数を減らしても同様に $p$ や $q$ の値によって効果が発揮する符号化は変わってくると考えられる。

また、用意するパケットの数が増えると全体的なDEPが上がってくることがわかった。XOR符号化の際にXOR演算しないパケットの数を固定し、用意するパケットの数を増やすと、復号が必要なパケットの数も増える。よって、復号で受信パケットを用いてXOR演算する回数が多くなるため、パケットに含まれたBFEの影響を受けやすくなったからだと考えられる。

### 5.3 各タイプに必要な消費電力の見積もり

第3章において、BCH(63,51)符号化、51-bit XOR符号化及び63-bit XOR符号化の消費電力の見積もりを行った結果は、表5.4の通りである。

表 5.4: 1回の符号化に必要な消費電力

	BCH(63,51)	51-bit XOR	63-bit XOR
Power[mW/clock cycle]	0.74	0.04	0.05

また、前節までに検証してきた各タイプにおいて用いた符号化の種類と回数は方式の表から得ることができる。ここでは、用意するパケットの数を9個とした事例に注目する。

タイプ1: BCH(63,51)符号化を9回

タイプ2: BCH(63,51)符号化を9回の後、63-bit XOR符号化を15回

タイプ3: 51-bit XOR符号化を9回の後、BCH(63,51)符号化を18回

以上より、表5.4の結果を用いると各タイプに必要な消費電力を見積もることができる。例えばタイプ2の場合、BCH(63,51)符号化は9回行われるので、ここで必要な消費電力は $0.74 \times 9 = 6.66[\text{mW}]$ となる。この後に63-bit XOR符号化が15回行われるので、ここで必要な消費電力

は  $0.005 \times 15 = 0.75[\text{mW}]$  となる．故にこれら2つの結果を足しあわせた  $6.66 + 0.75 = 7.41[\text{mW}]$  がタイプ2で必要な消費電力の見積もりになる．各タイプについても同様に見積もると表5.5の結果を得る．

表 5.5: 各タイプの消費電力見積もり

	BCH(63,51)	51-bit XOR	63-bit XOR	Total Power
タイプ1	6.66	-	-	6.66
タイプ2	6.66	-	0.75	7.41
タイプ3	13.32	0.60	-	13.92

これらの結果を見ると，タイプ3はタイプ1とタイプ2に比べておおよそ2倍の消費電力がかかることがわかる．これは，タイプ3はタイプ1やタイプ2より消費電力が大きいBCH(63,51)符号化を多く用いていることが原因である．

ここで，前の検証で得られた図5.5から図5.10を見てみると， $p$ が0.001と小さい時はタイプ2とタイプ3のDEPがほぼ同様であった．また，先ほどの見積もりで，タイプ2の消費電力はタイプ3の消費電力のおおよそ半分であることがわかっている．つまり， $p = 0.001$ のようにBFEが低い環境ではタイプ2はタイプ3の時と同様のDEPを得ることができ，かつ消費電力がタイプ3より少ない状態で用いることができる．故に，BFEが発生しにくい環境で行う通信には，タイプ2の符号化方式を用いるのが良いと考えられる．

## 5.4 復号できなかったパケットの個数

これまでの符号化に対する性能評価では，送るべきパケットが全て復号できたかどうかで評価してきた．本節では，より詳しく符号化の性能を評価するために，先ほどまでの検証で用いた各タイプにおいて，復号しても得られなかったパケットの数を調査した．今回は，5.1節で行った検証の環境において，復号できなかったパケット  $P_i$  の個数を求めた．さらに，試行を複数回行い，各試行で復号できなかった  $P_i$  の個数から次の「復号できなかったパケットの平均個数 (Average of Non Decoded Packets, ANDP)」を定義して，それを評価の指標に用いた．

### 定義 3

$$\text{ANDP} = \frac{\text{各試行で得られた復号できなかった } P_i \text{ の総数}}{\text{試行回数}}$$

$p$  は 0 から 0.50 まで 0.10 刻み, 場所によっては 0.01 刻みで変動させた. また,  $q$  は 0.001 から 0.006 まで 0.001 刻みで変動させた. さらに, 試行回数は 10 万回とした.

以上を加味すると, 各  $q$  における  $p$  と ANDP の関係は図 5.29 から図 5.34 までとなった.

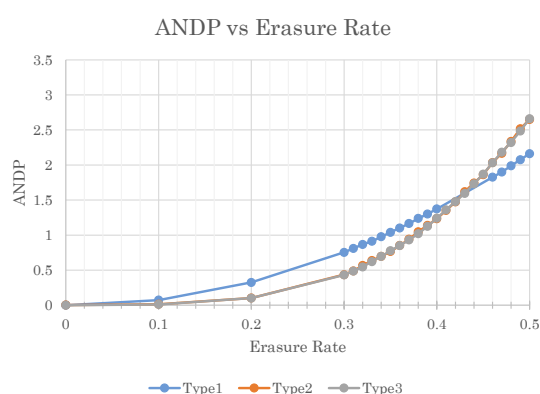


図 5.29:  $p = 0.001$

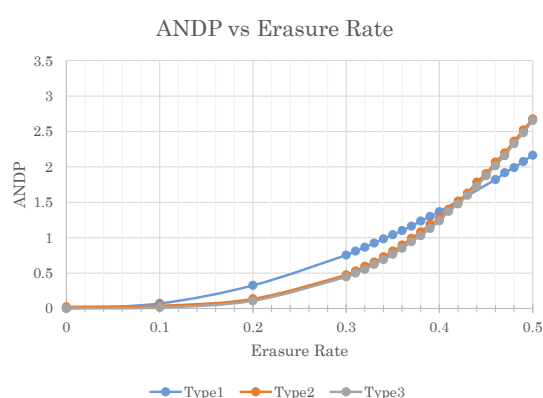


図 5.30:  $p = 0.002$

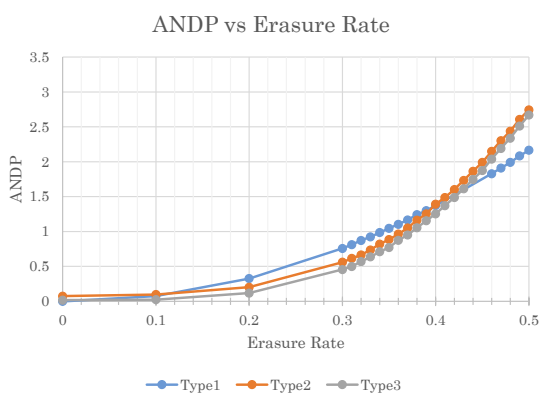


図 5.31:  $p = 0.003$

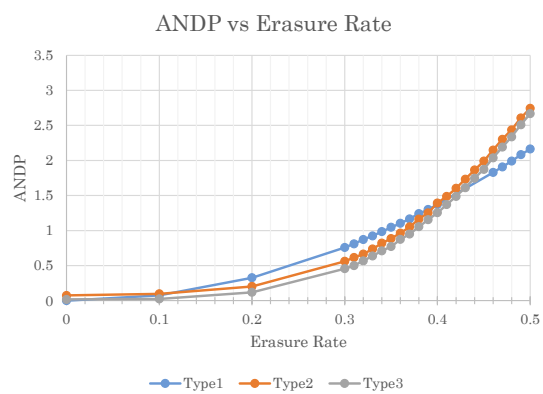


図 5.32:  $p = 0.004$

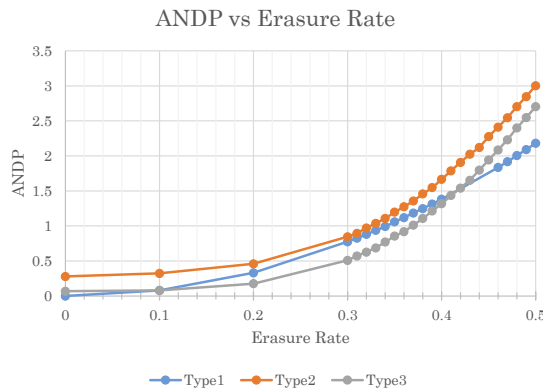


図 5.33:  $p = 0.005$

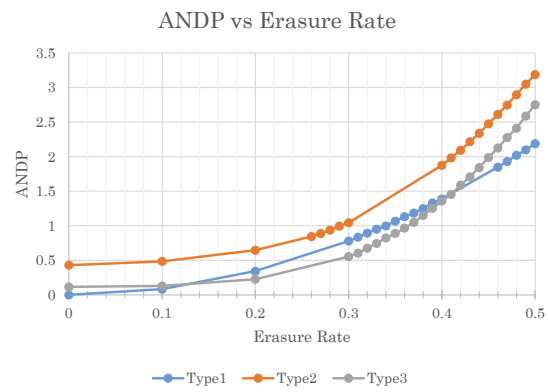


図 5.34:  $p = 0.006$

図5.29から図5.34までの結果より、 $q$ や $p$ を変化させることによりタイプごとに復号できなかったパケットの個数には、以下の特徴があることがわかった。

#### タイプ1

$p$ が0.001から0.006の間であれば、各 $q$ とANDPとの関係性は $p$ の値によらずほぼ変わらなかった。つまり、復号できなかったパケットは $p$ の値によらず次のように見積もることができる。

- $q < 0.35$ の時、およそANDPは1を下回るので復号できなかったパケットは0個
- $0.35 \leq q < 0.48$ の時、およそANDPは1以上となるが2を下回るので復号できなかったパケットは1個
- $0.48 \leq q$ の時、およそANDPは2以上となるが3を下回るので復号できなかったパケットは2個

#### タイプ2

タイプ1と異なり、 $p$ が0.001から0.006の間で各 $q$ とANDPとの関係性は $p$ の値によって大きく変化した。例えば、 $p = 0.001$ では図5.29より復号できなかったパケットは次のように見積もることができる。

- $q < 0.38$ の時、およそANDPは1を下回るので復号できなかったパケットは0個
- $0.38 \leq q < 0.46$ の時、およそANDPは1以上となるが2を下回るので復号できなかったパケットは1個

- $0.46 \leq q$  の時, おおよそ ANDP は 2 以上となるが 3 を下回るので復号できなかったパケットは 2 個

また,  $p = 0.006$  では図 5.34 より復号できなかったパケットは次のように見積もることができる.

- $q < 0.30$  の時, おおよそ ANDP は 1 を下回るので復号できなかったパケットは 0 個
- $0.30 \leq q < 0.42$  の時, おおよそ ANDP は 1 以上となるが 2 を下回るので復号できなかったパケットは 1 個
- $0.42 \leq q < 0.49$  の時, おおよそ ANDP は 2 以上となるが 3 を下回るので復号できなかったパケットは 2 個
- $0.49 \leq q$  の時, おおよそ ANDP は 3 以上となるが 4 を下回るので復号できなかったパケットは 3 個

よって, 各  $p$  において復号できなかったパケットの数を見積もる必要がある.  
タイプ 3

$p$  が 0.001 から 0.006 の間であれば, タイプ 1 と同様に各  $q$  と ANDP との関係性は  $p$  の値によらずほぼ変わらなかった. つまり, 復号できなかったパケットは  $p$  の値によらず次のように見積もることができる.

- $q < 0.38$  の時, おおよそ ANDP は 1 を下回るので復号できなかったパケットは 0 個
- $0.38 \leq q < 0.46$  の時, おおよそ ANDP は 1 以上となるが 2 を下回るので復号できなかったパケットは 1 個
- $0.46 \leq q$  の時, おおよそ ANDP は 2 以上となるが 3 を下回るので復号できなかったパケットは 2 個

復号できなかったパケットの数が少なければ少ないほど誤り訂正能力が高いといえるので, 以上の特徴をまとめると次のようなことがわかった.

$p = 0.001$  の時

- $0.35 \leq q < 0.38$  の時，タイプ1は修復できなかったパケットが1個生じるが，タイプ2とタイプ3は修復できなかったパケットの個数は0個であった．
- $0.46 \leq q < 0.48$  の時，タイプ2とタイプ3は修復できなかったパケットが2個生じるが，タイプ1は修復できなかったパケットの個数は1個であった．
- 上記の2つの場合を除く  $q$  の時は，各タイプにおける修復できなかったパケットの個数は同じであった．

よって， $q < 0.38$  の時はタイプ2とタイプ3の誤り訂正能力がタイプ1と比べて高いが， $0.46 \leq q$  の時は逆にタイプ1の誤り訂正能力がタイプ2とタイプ3と比べて高くなることがわかった．

$p = 0.006$  の時

- $0.30 \leq q < 0.35$  の時，タイプ2は復号できなかったパケットが1個生じるが，タイプ1とタイプ3は修復できなかったパケットの個数は0個であった．
- $0.35 \leq q < 0.38$  の時，タイプ1とタイプ2は修復できなかったパケットが1個生じるが，タイプ3は修復できなかったパケットの個数は0個であった．
- $0.42 \leq q < 0.46$  の時，タイプ2は復号できなかったパケットが2個生じるが，タイプ1とタイプ3は修復できなかったパケットの個数は1個であった．
- $0.46 \leq q < 0.48$  の時，タイプ2とタイプ3は修復できなかったパケットが2個生じるが，タイプ1は修復できなかったパケットの個数は1個であった．
- $0.49 \leq q$  の時，タイプ2は復号できなかったパケットが3個生じるが，タイプ1とタイプ3は修復できなかったパケットの個数は2個であった．
- 上記の2つの場合を除く  $q$  の時は，各タイプにおける修復できなかったパケットの個数は同じであった．

よって、 $q < 0.38$ の時はタイプ3の誤り訂正能力がタイプ1とタイプ2と比べて高いが、 $0.46 \leq q$ の時は逆にタイプ1の誤り訂正能力がタイプ2とタイプ3と比べて高くなることがわかった。

故に、復号できなかったパケットの個数から符号化の性能を評価したところ、 $q < 0.38$ の時はXOR符号化とBCH符号化を接続した接続符号化の方がBCH符号化単体よりも誤り訂正能力が高いことがわかるが、 $0.46 \leq q$ の時は逆にBCH符号化単体の方が接続符号化よりも誤り訂正能力が高いことがわかった。

前節の送るべきパケットが全て復号できたかどうかで符号化の性能を評価した時は、 $q$ が高い時は ( $0.2 \leq q \leq 0.5$ の範囲では)、常に接続符号化の方がBCH符号単体よりも誤り訂正能力が高いことがわかった。しかし、復号できなかったパケットの個数から符号化の性能を評価した時は、 $q$ が高い時に ( $0.46 \leq q \leq 0.5$ の範囲で)、BCH符号化単体の方が接続符号化よりも誤り訂正能力が高くなることがわかった。つまり、誤り訂正能力は送るべきパケットが全て復号できたかどうかで符号化の性能を評価するか、もしくは復号できなかったパケットの個数から符号化の性能を評価するかによって、XOR符号化とBCH符号化を接続させる方が良いか、それともBCH符号化単体を用いるほうが良いかは変わってくると言える。

## 5.5 再送方式の検討

実際の通信では、送られるはずのパケットがMSで復号を行っても得られなかった場合、得られなかったパケットを送信機から再送させる再送要求がある。この再送要求が生じた場合、再送要求が生じなかった場合と比べてパケットの再送による通信の遅延が生じたり、再送対象のパケットを再び符号化することによる電力消費が生じる。

これまでの検証では、各 $p$ や $q$ におけるDEP、及び符号化に必要な消費電力からどのタイプを用いることが適切なのかを検証した。しかし、実際の通信で用いることを考えると、更に再送についても加味する必要がある。再送について考えるには、各タイプにおいて復号しても得られなかったパケットの個数を調査する必要がある。その上で、全てのパケットが復号できるようにするための再送方式をそれぞれ検討して、各再送方式にかかる計算量や消費電力などの手間を見積もることが求められる。

この見積りには，前節で得られた「復号できなかったパケットの個数」の結果を用いることができるが，具体的な方式の検討については，今後の課題としたい．



## 第6章 まとめと今後の課題

### 6.1 まとめ

本研究では、WBANのための消費電力が少なく、かつ通信誤りにも強い符号化として、XOR演算を用いた符号化であるXOR符号化とBCH符号の符号化であるBCH符号化を接続させた接続符号化の提案・検証を行った。

まず、提案する接続符号化を構成するXOR符号化とBCH符号化において、それぞれの符号化に必要な消費電力の見積りを試みた。XOR符号化は、51-bit同士のデータをXOR演算する51-bit XOR符号化、及び63-bit同士のデータをXOR演算する63-bit XOR符号化とした。さらにBCH符号化は、IEEEがWBANへ用いることを推奨しているBCH(63,51)符号化[8]とした。それぞれ3つの符号化の消費電力を、FPGA設計を用いることによって見積もりを行った。

次に、様々なXOR符号化方式についてパケット損失に対する耐性の検証・比較を行った。始めにセンサーの数を9、冗長度は2とした時に、パケット損失に対する耐性を既存研究[7]のアルゴリズムから構成された符号化の方式について検証した。さらに、センサーの数や冗長度を固定した状態で幾つか他の方式についても検証したが、既存研究[7]のアルゴリズムから構成されたものより高いパケット損失に対する耐性があるものを見つけることができなかった。また、冗長度は2に固定し、センサーの数を4,5,6にした時に、パケット損失に対する耐性が最も高くなる方式を見つけた。さらに、各センサーの数における既存研究[7]のアルゴリズムから構成された符号化方式と、パケット損失に対する耐性が最も高くなる方式を比較したところ、必ず一致するとは限らないことがわかった。一方で、計算機の性能の関係からセンサーの数を7以上とした時にパケット損失に対する耐性が最も高くなる方式の検証をすることが出来なかった。しかしながら、同様にXOR演算しないパケットの数が3または4の時に、耐性が最も高くなるような方式が存在すると予想される。

さらに、提案する接続符号化において、パケット損失とパケット内のビット反転誤り (BFE) が同時に発生する場合に対する耐性の検証を行った。今回は BCH(63,51) 符号化のみを行う符号化をタイプ1, BCH(63,51) 符号化の後に XOR 符号化を行う符号化をタイプ2, XOR 符号化の後に BCH(63,51) 符号化を行う符号化をタイプ3と名付けて、各タイプの検証を行った。その結果、パケット損失が発生する確率や BFE が発生する確率によって復号誤り確率 (DEP) が最も低くなるタイプは異なることがわかった。その中の特徴の1つとして、BFE が発生する確率が 0.001 と低い場合、タイプ2とタイプ3はパケット損失が発生する確率に関わらずほぼ同じ DEP で、かつタイプ1より高くなることがわかった。また、タイプ3は BCH(63,51) 符号化の回数が他のタイプより多いため、消費電力は他と比べておよそ2倍になることがわかった。よって、BFE が発生する確率が低い環境下ではタイプ2の符号化を用いるのが良いことがわかった。

また、ここまでの検証では、送るべきパケットが全て復号できたかどうかから誤り訂正能力を評価していたが、さらに詳しく検証するために、復号で得られなかった送るべきパケットの個数を調査した。その結果、パケット損失が発生する確率が 0.46 以上になる時、接続符号化より BCH 符号化単体の方が復号で得られなかったパケットの個数が少なくなることがわかった。つまり、符号化の誤り訂正能力を評価する時、「送るべきパケットが全て復号できたかどうか」と「復号で得られなかった送るべきパケットの個数」のどちらかを選ぶかにより最適な符号化の判断が異なることがわかる。

## 6.2 今後の課題

今後の課題として、次の4点が挙げられる。

- XOR 符号化方式について、第4章にてセンサーの数を 4,5,6 個とした時に誤り耐性が最も高くなる方式を見つけた検証結果を用いて詳細な理論的解析を行い、センサーの数が7個以上の場合でも最も誤り耐性が高くなるような方式を検討する。
- 実際の通信では、MS 側で復号できなかったパケットを送信側から再送させる再送要求が行われる。本研究では各タイプについて誤りや消費電力の観点から検証を行ったが、実際の現場で用いるには再送

要求が発生した際の計算量や消費電力などの手間も考える必要がある。第5章において各タイプが通信を行った時に復号できなかったパケットの個数を調査したので、この結果を考慮して各タイプにおける再送方式の検討を行う。さらに、得られた各タイプの再送方式から必要な計算量や消費電力などを加味して、実際に用いるにはどのタイプが適切なのかを検証する。

- 生体データは個人情報であるため、流出を嫌う人達が出てくると考えられる。よって、暗号化や秘密分散などを行うことによって機密性を維持できるように検討する。
- 提案した符号化を実際のWBANで通信を行う実機に組み込むことにより、実際に無線通信を行った時の性能を調査する。

# 謝辞

本研究の実施および本論文の作成にあたり、2年間多くの御指導，御助言を下された森田啓義先生，眞田亜紀子先生に心から感謝いたします。また，ゼミ等で様々な御助言を下された応用ネットワーク学講座の笠井裕之先生，研究生活を支えて下さった秘書の片桐真喜子さん，ならびにお世話になりました森田研究室，笠井研究室の学生の皆さまに感謝いたします。さらに，インターンシップでお世話になり，私の研究にも様々な御助言を下された富士通研究所の井田一郎様を始め，富士通研究所先端ワイヤレス研究部の皆さまに感謝いたします。

## 参考文献

- [1] 内閣府, “平成 25 年度 高齢社会白書,” 2013 年 6 月.
- [2] 朽久保修, and 山末耕太郎. “3. 情報通信技術を活用したヘルスケアネットワークシステム (<小特集> 情報通信技術によるユビキタス医療への挑戦).” 電子情報通信学会誌 90.8 (2007): 636-641.
- [3] 菅野宏斗, et al. “B-20-30 ワイヤレス生体センサを用いたユビキタス健康監視に向けた臨床試験 (1): ホルター心電計とのデータ比較 (B-20. ユビキタス・センサネットワーク, 一般セッション).” 電子情報通信学会総合大会講演論文集 2 (2010): 614.
- [4] 小崎隆明, et al. “B-20-31 ワイヤレス生体センサを用いたユビキタス健康監視に向けた臨床試験 (2): 3 軸加速度データによる行動解析 (B-20. ユビキタス・センサネットワーク, 一般セッション).” 電子情報通信学会総合大会講演論文集 2 (2010): 615.
- [5] 幸島明男, and 車谷浩一. “携帯電話を用いたモバイル生体センサデータ解析プラットフォームの構築.” 情報処理学会論文誌 51.4 (2010): 1192-1203.
- [6] S. Marinkovic and E. Popovici, “Network Coding for Efficient Error Recovery in Wireless Sensor Networks for Medical Applications,” 2009 First Int. Conf. on Emerging Network Intelligence, pp. 15-20, 2009.
- [7] E. Byrne, A. Manada, S. Marinkovic, E. M. Popovici, “A graph theoretical approach for network coding in wireless body area networks,” Proceeding of the IEEE International Symposium on Information Theory (ISIT2011), pp. 405-409.
- [8] LAN/MAN Standards Committee, “IEEE Standard for Local and metropolitan area networks Part15.6: Wireless Body Area Networks,” <http://standards.ieee.org/getieee802/download/802.15.6-2012.pdf>.

- [9] N. F. Timmons, and W. G. Scanlon. “An adaptive energy efficient MAC protocol for the medical body area network.” *Wireless Communication, Vehicular Technology, Information Theory and Aerospace & Electronic Systems Technology*, 2009. Wireless VITAE 2009. 1st International Conference on. IEEE, 2009.
- [10] K. Yokota, A. Manada, and H. Morita, “An XOR encoding for wireless body area networks,” *Proceedings of the 8th International Conference on Body Area Networks, ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering)*, pp. 240-243, 2013.